FITTING SURFACES TO SCATTERED DATA

Larry L. Schumaker

This paper is a survey of a variety of numerical methods for fitting a function to data given at a set of points scattered throughout a domain in the plane. We discuss four classes of methods: (1) global interpolation, (2) local interpolation, (3) global approximation, and (4) local approximation. We also discuss two-stage methods and contouring. The surfaces constructed will include polynomials, spline functions, and rational functions, among others.

1. Introduction

Our aim is to survey methods for solving the following problem.

PROBLEM 1.1. Let D be a domain in the (x,y)-plane, and suppose F is a real-valued function defined on D. Suppose we are given the values $F_i = F(x_i,y_i)$ of F at some set of points (x_i,y_i) located in D, $i = 1,2,\ldots,N$. Find a function f defined on D which reasonably approximates F.

This problem is, of course, precisely the problem of fitting a surface to given data. In many cases the domain D is a rectangle and the data points lie on a rectangular grid. There are, however, many practical problems (see the following section for some specific examples), where D is of unusual shape and where the data points are irregularly scattered throughout D. Thus, while we shall pay some attention to special methods for regularly spaced data, we are actually more interested in the general case.

There are basically two approaches to handling Problem 1.1. First, we may try to construct a function f which interpolates

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER (18) AFOSR-TR-76-0872 (19)	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) (6) FITTING SURFACES TO SCATTERED DATA,	5. TYPE OF REPORT & PERIOD COVERED (9) Interim Rept.,
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) Larry L. Schumaker	8. CONTRACT OR GRANT NUMBER(s) (15) AF-AFOSR-2598-74
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Texas Department of Mathematics Austin Texas 78712	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 9749-03
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research (NM) Bolling AFB, DC 20332	12. REPORT DATE (11) 1976 (12) 67p. 13. NUMBER OF PAGES 66
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) (16) AF-9749 (17) 9749-03	15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

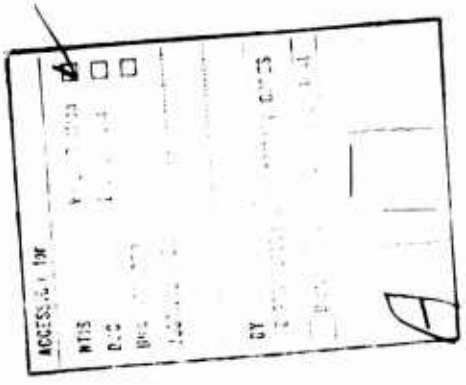
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Surface fitting
Spline functions
Interpolation
Data fitting

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper is a survey of methods for fitting a surface to a set of data scattered throughout a plane region. The methods surveyed include interpolation, smoothing, least-squares, and a variety of other direct and approximation methods. The fitting functions include polynomials, rationals, splines, and others. Numerical methods are emphasized, and contouring is also discussed.

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE 241823 UNCLASSIFIED



the data exactly; i.e., such that

(1.1) $f(x_i,y_i) = F_i, \quad i = 1,2,\ldots,N.$

This approach may be desirable when the function values at the data points are known to high precision and where it is highly desirable that these values be preserved by the approximating function.

The second approach involves constructing f which only approximately fits the data. This may be regarded as data smoothing and will be desirable when (as is often the case) the data are subject to inaccurate measurement or even errors. The question of whether interpolation or approximation should be used will not be discussed further here--this is a problem which must be settled for the individual problem at hand.

In discussing Problem 1.1, it will be convenient to make a further distinction between those methods which are local in character (i.e., where the value of the constructed surface f at the point (x,y) depends only on the data at relatively nearby points) and those methods which are global in nature. Thus, we discuss four categories of methods in sections 3-6: (1) global interpolation, (2) local interpolation, (3) global approximation, and (4) local approximation. In each of these sections we further subdivide the material according to the type of functions being used and the type of data (scattered or not) for which the method is suitable.

In discussing methods which apply only to special arrangements of data points, we have two objectives in mind. First, the methods are of interest in their own right. More importantly in terms of Problem 1.1, however, such methods can also be used in two-stage processes in which we first construct a surface g based on the scattered data, and then use g to generate regular data for the construction of another (perhaps smoother or more convenient) surface f. Such two-stage methods

will be discussed (along with several examples) in more detail in section 7.

For many of the methods based on regular data and some of those for scattered data, error bounds are available to indicate how well smooth functions are approximated by the surface constructed. We do not have space to go into the extensive literature on error bounds. A simple test of how well a method will approximate smooth functions is, however, provided by its ability to reproduce polynomial surfaces exactly (that is, if F is a polynomial in x and y up to a certain degree, then the surface f is identically equal to F). For many of the methods we will be able to indicate the corresponding degree of exactness.

In many of the applications of surface-fitting techniques (cf. the examples in section 2), the ultimate aim is to use the data to construct a contour map of the unknown function. Since F is known only at the data points, we must be content to construct a contour map for one of our fitted surfaces. In section 8 we discuss some approaches to accomplishing this numerically.

We close this introduction with a disclaimer--this survey does not include all possible methods for fitting surfaces to scattered data. For example, we have not discussed Fourier series methods, spatial filtering, and other such related statistical techniques. In addition, the set of references for those methods which we have discussed are also not complete. My original intention was to compile as complete a bibliography as possible, but the sheer bulk of relevant papers and my inability to locate all of them convinced me to settle for less. I have opted to quote a fairly representative list of papers, including several other surveys. Further references can be found by consulting these. I shall be very happy to receive information on references and methods I have overlooked.

2. Examples

In this section we shall quote several explicit examples of Problem 1.1 to emphasize the fact that unusually shaped regions and scattered data do arise frequently in practice.

EXAMPLE 2.1. Petroleum exploration. In exploring for petroleum, the contours of various underground layers of sandstone, shale, limestone, etc. can be important indicators of possible oil fields. Frequently, data on such layers is available from exploratory wells, which, however, have most likely been drilled at locations scattered randomly throughout some geographical region of interest. To quote a specific example, Robinson, Charlesworth, and Ellis [166] consider precisely this problem for some data obtained from 7,500 wells drilled in Alberta. For another example of this type, see Whitten and Koelling [208].

Problems similar to that mentioned in Example 2.1 arise frequently in cartography and submarine topography where the measurements represent actual elevations. In some cases the measurements must be taken from photographs or from sonar measurements and are usually subject to some measurement error (eg. see Kubik [125] for a discussion of photogrammetry).

EXAMPLE 2.2. Geological maps. There are a great many problems in Geology and the earth sciences in which the data arises from some other function of location besides actual elevations. For example, some geological variables of interest might include concentrations of various chemicals, specific gravity, electrical resistivity, grain size, texture, optical properties, isotope ratios, etc. To quote a specific example, Bhattacharyya [21, 22] discusses methods for fitting a surface to measurements (taken by airborne sensors) of magnetic potentials over a certain portion of the Yukon. See also Bhattacharyya and Raychaudhuri [23] and Crain and Bhattacharyya [61].

The importance of surface-fitting methods in the earth sciences can be judged by the large number of papers in the area relating to various fitting methods. For a further list of problems and a discussion of some of the methods which have been applied, see the books of Bohrenberg and Giese [31], Chorley [51], David [62], Harbaugh and Merriam [98], and Merriam [140]. Recent survey papers include Whitten [203, 205] and Whitten and Koelling [207]. To add just a few more of the papers in the geological literature dealing with surface fitting to our list, we mention Anderson [7], Grant [91], Hessing, Lee, and Pierce [114], Holroyd and Bhattacharyya [115], Kubik [123, 125], Norcliffe [151], Reilly [162], Whitten [200, 201, 204], and Whitten and Koelling [206].

EXAMPLE 2.3. Heart potentials. In order to diagnose certain abnormal heart conditions, it is desired to make a series of several hundred contour maps of the heart potential field at time steps of 1/100 of a second throughout a heart beat. Data on these heart potentials can be obtained by fitting the patient with a shirt containing probes. Because of body geometry, when this shirt is flattened out it takes the nonrectangular form illustrated in Figure 1. Although the probes could be arranged fairly regularly in this domain, because of the added signifi-

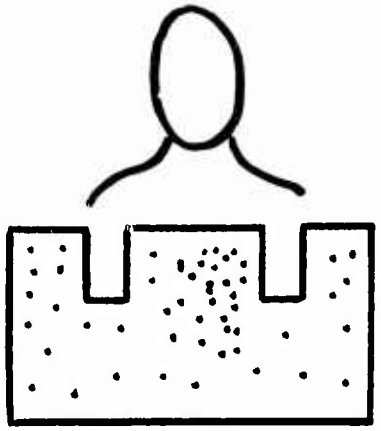


Figure 1. Heart Potential Measurements

cance of frontal measurements, in practice more probes are fitted there than in the back. This example was brought to my attention by Ms. Patrizia Ciarlini of Rome.

Potential fields arise in many other applications. We have already mentioned Geology in Example 2.2. For some examples in modelling plasmas see Buneman [40]. The problem arises in Biersack and Fink [24] in experimentally studying crystal structure using neutron bombardment. Data from waveform distortion in electronic circuits can be found in Akima [5, 6].

3. Global interpolation methods

In this section we outline several methods for solving the interpolation problem (1.1).

3.1 Polynomial interpolation. (Scattered data). The general theory of finite dimensional interpolation is, of course, very well known (e.g., see Davis [63]). Briefly, if $\{\phi_i\}_1^N$ are N functions defined on the domain D, then the function

$$(3.1) \quad f(x,y) = \sum_{j=1}^{N} a_j \phi_j(x,y)$$

will satisfy (1.1) if and only if $\{a_j\}_1^N$ is a solution of the linear system

$$(3.2) \quad \sum_{j=1}^{n} a_j \phi_j(x_i,y_i) = F_i, \quad i = 1,2,\ldots,N.$$

This system has a (unique) solution for arbitrary choices of data precisely when it is nonsingular. This depends on the choice of functions $\{\phi_j\}_1^N$ and the location of the data points.

To illustrate this method, we may choose the $\{\phi_j\}_1^N$ to be polynomials in x and y. Given N, there is some leeway in the choice of which powers of x and y to use. For example, with N = 3 one could use the functions 1, x, y or possibly the functions 1, x^2, y^2, etc. When N is of the form N =

$(d+1)(d+1)$, we might use the functions

$$\{\phi_j(x,y)\}_1^N = \{x^\nu y^\mu\}_{\nu=0,\mu=0}^{d,d}$$

As simple as this sounds, there are some serious difficulties with polynomial interpolation of scattered data. For openers, it is not so easy to guarantee that the system (3.2) is nonsingular. To give a very simple example, consider the case $N = 3$ with the functions $1, x, y$. If the three data points happen to lie on a line, then (3.2) will in fact be singular. Even when (3.2) is nonsingular, it will often be the case (at least if N is moderately large) that the system will be ill-conditioned. Finally, as is well known, polynomials of even moderate degree exhibit a considerable oscillatory character, and the resulting surface (even though it is C^∞) is often too undulating to be acceptable. The general problem of polynomial interpolation to scattered data is not usually treated in Numerical Analysis and Approximation Theory books (see, however, Kunz [126], Prenter [157], and Steffenson [186]). Some papers dealing with the question include Guenther [93], Thatcher [189], Thatcher and Milne [190], and Whaples [197]. Assuming the interpolant exists, error bounds have been studied in Ciarlet and Raviart [52-55].

Let

$$(3.3)\quad \mathcal{P}_{m,n} = \text{span}\,\{x^\nu y^\mu\}_{\nu=0,\mu=0}^{m,n}$$

be the space of polynomials of degree m in x and of degree n in y. This linear space is of dimension $(m+1)(n+1)$ and is, in fact, the tensor product of the linear spaces $\mathcal{P}_m$ and $\mathcal{P}_n$. It is perhaps of interest to note that there always exists a (usually nonunique) polynomial $p \in \mathcal{P}_{N,N}$ which solves the interpolation problem (1.1), no matter how the data points are positioned, see Prenter [158].

3.2 Polynomial interpolation (gridded data). We begin this subsection by defining what we mean by gridded data. Let

$$H = [a,b] \times [c,d] \tag{3.4}$$

be a rectangle, and let

$$\begin{aligned} a = x_0 < x_1 < \ldots\ldots < x_{k+1} = b \\ c = y_0 < y_1 < \ldots\ldots < y_{\ell+1} = d. \end{aligned} \tag{3.5}$$

We suppose now that F is a function defined on H, and that we have the values of F at the corner points of the rectangular grid defined by (3.5); i.e.,

$$F_{ij} = F(x_i, y_j), \quad \begin{matrix} i = 0,1,\ldots,k+1 \\ j = 0,1,\ldots,\ell+1. \end{matrix} \tag{3.6}$$

This is a total of $N = (k+2)(\ell+2)$ data points.

It is quite easy to show that there exists a unique polynomial p in the class $\mathcal{P}_{k+1,\ell+1}$ (cf. the definition (3.3)) which interpolates the gridded data given in (3.4)-(3.6). In fact, p can be written down explicitly in terms of the one-dimensional Lagrange polynomials as

$$p(x,y) = \sum_{i=0}^{k+1} \sum_{j=0}^{\ell+1} F_{ij} L_i(x) \tilde{L}_j(y), \tag{3.7}$$

where the $\{L_i(x)\}_0^{k+1}$ and $\{\tilde{L}_j(y)\}_0^{\ell+1}$ are the usual one-dimensional Lagrange polynomials associated with the interpolation points $\{x_i\}_0^{k+1}$ and $\{y_j\}_0^{k+1}$, respectively. Interpolation of gridded data by polynomials has been discussed in various books and papers--we do not bother with a long list. See e.g. Prenter [157] or Steffenson [186]. More recently, there has been considerable work on Hermite and osculatory interpolation in several variables; see e.g. Ahlin [3], Haussman [99,101,102], and Salzer [168-170].

3.3 Shepard's method. In this subsection we discuss a method of Shepard [180] and some modifications of it. The method applies to arbitrarily spaced data, and the interpolating function can be written down explicitly.

Let ρ be some metric in the plane, for example the usual distance metric. Given a point (x,y), let $r_i = \rho((x,y),(x_i,y_i))$ for $i = 1,2,\ldots,N$. Let $0 < \mu < \infty$. Then Shepard's interpolation formula is defined by

$$(3.8)\quad f(x,y) = \begin{cases} \left(\sum_{i=1}^{N} \frac{F_i}{r_i^{\mu}}\right) \Big/ \left(\sum_{i=1}^{N} \frac{1}{r_i^{\mu}}\right), & \text{when } r_i \neq 0, \text{ all } i \\ F_i, & \text{when } r_i = 0. \end{cases}$$

The formula (3.8) is defined for all points (x,y) in the plane R^2. It is clear from the definition that it interpolates the values F_i at the data points (x_i,y_i), $i = 1,2,\ldots,N$. The value of $f(x,y)$ at nondata points is obtained as a weighted average of all the data values, where the i^{th} measurement is weighted according to the distance of (x,y) from the point (x_i,y_i).

We shall briefly recount some of the properties of Shepard's formula. First, by converting all of the terms to a common denominator, it can be shown that

$$(3.9)\quad f(x,y) = \sum_{i=1}^{N} F_i A_i(x,y),$$

where

$$(3.10)\quad A_i(x,y) = \frac{\prod_{\substack{j=1 \\ j\neq i}}^{N} [r_j(x,y)]^{\mu}}{\sum_{k=1}^{N} \sum_{\substack{\ell=1 \\ \ell\neq k}}^{N} [r_\ell(x,y)]^{\mu}}, \quad i = 1,2,\ldots,N.$$

These functions satisfy

(3.11) $A_i(x_j,y_j) = \delta_{ij}, \quad i,j = 1,2,\ldots,N.$

The representation (3.9) is numerically more stable than the original formula (3.8).

In view of its definition, we see that the function $f(x,y)$ constructed by Shepard is not a simple polynomial or rational function. It is clear, however, that except for the points (x_i,y_i), it is analytic everywhere in the plane. Its behavior in the vicinity of the data points (x_i,y_i) depends on the size of μ. It can be shown that for $0 < \mu \leq 1$, f has cusps at these points. For $1 < \mu$, f has flat spots at the data points (i.e., the partial derivatives vanish there). We also observe the interesting property that

(3.12) $$\min_{1\leq i\leq N} F_i \leq f(x,y) \leq \max_{1\leq i\leq N} F_i.$$

We may also note that if the data came from a constant function, i.e., $F_i = c$, $i = 1,2,\ldots,N$, then f is also the constant function $f \equiv c$.

We now comment on the choice of μ. To get smooth surfaces without cusps, it is desirable to take $1 < \mu$. On the other hand, if μ is relatively large, then the surface tends to become very flat near the data points and consequently quite steep at points in between. Experiments (cf. Gordon and Wixom [90], Poeppelmeir [155], and Shepard [180]) seem to indicate that a choice of $\mu = 2$ is perhaps a good tradeoff. ([155] contains several examples showing the behavior as a function of μ.)

There are several drawbacks to Shepard's method (3.8), as pointed out by Shepard [180] himself. First, if N is large, then there is a very considerable amount of calculation involved in evaluating $f(x,y)$ at a particular point. Secondly, the weights are assigned on the basis of the distance of points from (x,y) only, not their direction. Finally, the flat spots

in the neighborhood of the data points is somewhat disturbing. The first of these objections can be met by defining a local version of the formula, which we shall do in section 4.5. It is possible to construct an analogous formula which accounts for direction. For details, see Shepard [180]. Finally, we briefly discuss handling the flat spots.

Suppose in addition to the function values F_i at each point (x_i,y_i) we also have estimates FX_i and FY_i of $F_x(x_i,y_i)$ and $F_y(x_i,y_i)$. Then we may consider the function

$$(3.13)\quad f(x,y) = \sum_{i=1}^{N} A_i(x,y)\,[F_i + (x-x_i)FX_i + (y-y_i)FY_i].$$

It is easily checked that this function also interpolates, and that

$$(3.14)\quad f_x(x_i,y_i) = FX_i, \quad f_y(x_i,y_i) = FY_i, \quad i = 1,2,\ldots,N.$$

This property may be expressed in the assertion that if the data F_i, FX_i, FY_i came from a plane surface F, then f will exactly reproduce this surface. To use formula (3.13) in practice on the data-fitting Problem 1.1, we have to carry out a two-stage approximation process in which the first stage consists of some method for estimating the slope at each of the data points.

It might be of practical interest in some cases to construct still a more sophisticated version of Shepard's formula which would exactly reproduce higher-order polynomial surfaces. One approach to doing this is to use the following lemma.

LEMMA 3.1. (Barnhill [15]). Let P and Q be linear projections of some linear space of functions $\mathcal{F}$ into itself. Suppose that Q exactly reproduces the linear subspace $E \subset \mathcal{F}$; i.e.,

$$(3.15)\quad Qp = p, \quad \text{all} \quad p \in E.$$

In addition, suppose that $\{\lambda_i\}_1^m$ is a set of linear functionals on $\mathcal{F}$, and that

(3.16) $\lambda_i Pf = \lambda_i f$, all $f \in \mathcal{F}$, $i = 1,2,\ldots,m$.

Then the Boolean sum projector

(3.17) $P \oplus Q = P + Q - PQ$

enjoys the function precision of Q (i.e., reproduces E) and the interpolation properties of P (i,e., (3.16) also holds for $P \oplus Q$).

This result permits the construction of interpolation schemes using Shepard's formula which reproduce higher-order surfaces. For an example, see Poeppelmeir [155] where Shepard's formula is combined with a certain local interpolation scheme which reproduces quadratic surfaces. In closing this section we note that Shepard's formula can also be interpreted as arising from weighted least squares--see section 5.1.

3.4 Spline interpolation (scattered data). Suppose X is a linear space of "smooth" functions defined on the domain D, and let

(3.18) $U = \{f \in X: f(x_i,y_i) = F_i, \quad i = 1,2,\ldots,N\}$.

U is the set of smooth functions which interpolate. Now suppose that θ is a functional on X which measures the smoothness of an element in X--the smaller $\theta(f)$ is, the smoother f is. Then we may consider the following minimization problem:

(3.19) Find $s \in U$ such that $\theta(s) = \inf_{u \in U} \theta(u)$.

The function s will be the smoothest interpolant, and in view of the similarity with classical spline approximation, s is called a spline function interpolating F. The basic questions concerning spline interpolation center around existence, uniqueness, characterization, and construction. A quite general

abstract theory of spline interpolation has been built up (see eg. Laurent [127] and references therein). In this section we quote some specific examples which can be used on Problem 1.1.

Where X is a semi-Hilbert space, $\theta(f) = \|f\|$, where $\|\cdot\|$ is a seminorm on X, and $N = \{f \in X: \|f\| = 0\}$, it is possible to show (under some additional mild conditions on X, see Duchon [72, 73]) that problem (3.19) always has a solution which is unique up to an element in N. Moreover, it can be shown that there exists a reproducing kernel K defined on $D \times D$ such that

$$(3.20) \quad s(x,y) = \sum_{i=1}^{N} a_i K((x,y);(x_i,y_i)) + \sum_{i=1}^{d} b_i p_i(x,y),$$

where $\{p_i\}_1^d$ is a basis for N. Moreover, the coefficients $\{a_i\}$ and $\{b_i\}$ can be determined from the linear system of equations

$$(3.21) \quad \sum_{j=1}^{N} K((x_j,y_j);(x_i,y_i))a_i + \sum_{i=1}^{d} b_i p_i(x_j,y_j) = F_j, \quad j=1,\ldots,N$$

$$\sum_{i=1}^{N} a_i p_k(x_i,y_i) = 0, \quad k = 1,2,\ldots,d.$$

The development with semi-Hilbert spaces in Duchon [72, 73] is an extension of earlier work of Atteia [10-12] and Thomann [192-193] using Hilbert spaces. The essential difficulty in applying the general results is the construction of an appropriate reproducing kernel. We turn now to some specific examples.

Suppose X is the space of all functions on the rectangle $D = H$ (cf. (3.4)) which have (distributional) derivatives up to order 2 which lie in $L^2(H)$. For $f \in X$, let

$$(3.22) \quad \theta(f) = \iint_D |D_x^2 f|^2 + 2|D_x D_y f|^2 + |D_y^2 f|^2.$$

The reproducing kernel in this case can be written down as an infinite series involving sin and cos, and the space N is spanned by 1, x, and y. Similarly, if we replace H by the unit disc UD, the kernel can be computed as an infinite series (see Atteia [10-12] and Thomann [192-193]). Thomann considers computation of these splines by approximating the infinite series--FORTRAN programs are also included.

If we replace the bounded sets H or UD by the entire plane R^2 and introduce an appropriate space X, it is possible to obtain explicit expressions for the reproducing kernel. This is the content of Duchon [72, 73]. In particular, let $\tilde{H}^s$ be the set of all tempered distributions f on R^2 whose Fourier transforms $\hat{f}$ satisfy $\int |\hat{f}| t^{2s} dt < \infty$. Let X^{ms} denote the set of all functions which have derivatives up to order m lying in $\tilde{H}^s$. Our first example concerns the space X^{20}. If we choose θ as in (3.22), then the interpolating spline solution of (3.19) is of the form

$$(3.23) \quad s(x,y) = \sum_{i=1}^{N} a_i r_i^2(x,y) \log (r_i(x,y)) + b_1 x + b_2 y + b_3,$$

where $r_i(x,y) = [(x-x_i)^2 + (y-y_i)^2]^{\frac{1}{2}}$. The coefficients are determined from the system (3.21) with $d = 3$, $N = \text{span}\{1,x,y\}$, and $K(z,w) = |z-w|^2 \log(z-w|$. Duchon refers to this type of spline as a <u>thin plate spline</u> since the expression θ relates to the energy in a thin plate forced to interpolate the data. This spline belongs to $C(R^2)$.

As a second example, suppose we consider $X = X^{21}$. In this case the solution of (3.19) with θ given by (3.22) has the form

$$(3.24) \quad s(x,y) = \sum_{i=1}^{N} a_i (r_i(x,y))^3 + b_1 x + b_2 y + b_3.$$

Here $K(z,w) = |z-w|^3$. Duchon [72, 73] refers to these splines

as pseudo-cubic splines because of the analogy with the cubic splines in one variable. They belong to $C^1(R)$. Pseudo quintic splines etc. are also considered in Duchon [72, 73].

A similar program has been carried out by Mansfield [133-137] for some spaces of smooth functions defined on a rectangle H. In [136] she considers a space of functions $T^{m,n}(\alpha,\beta)$, where m and n are positive integers and $a \leq \alpha \leq b$, $c \leq \beta \leq d$. This space is actually defined by completion of a set of tensor product functions with respect to an appropriate inner-product, and we do not want to define it precisely here. A function $f \in T^{m,n}(\alpha,\beta)$ has the following properties, however:

$$(3.25)\begin{cases} f^{(i,j)} \in C(H), \quad i = 0,1,\ldots,m-1 \text{ and } j = 0,1,\ldots,n-1 \\ f^{(s-j-1,j)}(x,\beta) \in AC[a,b] \text{ and } f^{(s-j,j)}(x,\beta) \in L^2[a,b], \\ \qquad\qquad j = 0,1,\ldots,n-1 \\ f^{(i,s-i-1)}(\alpha,y) \in AC[c,d] \text{ and } f^{(i,s-i)}(\alpha,y) \in L^2[c,d], \\ \qquad\qquad i = 0,1,\ldots,m-1 \\ f^{(m-1,n-1)} \in AC(H) \quad \text{and} \quad f^{(m,n)} \in L^2(H), \end{cases}$$

where AC stands for the space of absolutely continuous functions and where $s = m+n$. By constructing an appropriate reproducing kernel, she is able to solve problem (3.19) with

$$(3.26)\quad \theta(f) = \iint_H [f^{(m,n)}]^2 + \sum_{j=0}^{n-1} \int_a^b [f^{(s-j,j)}(x,\beta)]^2 dx + \sum_{i=0}^{m-1} \int_c^d [f^{(i,s-i)}(\alpha,y)]^2 dy.$$

In [133], Mansfield carries out a similar analysis for a space of functions $R^{m,n}$ defined on the rectangle H. Here $R^{m,n} = L_2^m[a,b] \times L_2^n[c,d]$, where $L_2^m[a,b]$ is the usual Sobolev space of functions with absolutely continuous derivatives up to order m-1, and with $f^{(m)} \in L^2[a,b]$. By constructing an

appropriate reproducing kernel, she now solves problem (3.19) with

$$(3.27)\quad \theta(f) = \iint_H [f^{(m,n)}]^2 + \sum_{j=0}^{n-1} \int_a^b [f^{(m,j)}(x,c)]^2 dx + \sum_{i=0}^{m-1} \int_c^d [f^{(i,n)}(a,y)]^2 dy .$$

The solution turns out to be a piecewise polynomial of degree $2m-1$ in x and of degree $2n-1$ in y. It is also in $C^{2m-2,2n-2}(H)$. For the particular case of gridded data, it reduces to the tensor product of one-variable splines (cf. the following section). Other more general definitions of θ are also considered (with minor modifications on the one-dimensional integrals).

A more algebraic approach to constructing multidimensional spline functions (which also involves certain kernel functions) has been taken by Schaback [173-174]. His two-dimensional kernel function is obtained as a tensor product of one-dimensional kernels.

3.5. Spline interpolation (gridded data). The problem of constructing interpolating splines in two dimensions with gridded data as in (3.4)-(3.6) is, of course, a special case of the general problems discussed in subsection 3.4. The development of the gridded data case predated the more general development and, moreover, is considerably simpler. There are a great many papers on two-dimensional polynomial splines and generalizations. We do not have space here to discuss all of them in detail. We shall be content to quote some of the papers and to give a somewhat more complete discussion of polynomial splines, which are the most widely used splines for this problem.

Some early papers dealing with two-dimensional interpolating splines include Birkhoff and de Boor [26], Birkhoff and

Garabedian [27], Price and Simonson [159], and Theilheimer and Starkweather [191]. In [26] certain bicubic splines were introduced which were later studied in detail in de Boor [32]. The problem was to minimize

$$(3.28)\quad \int_a^b \int_c^d [f^{(2,2)}(x,y)]^2 dxdy$$

over all appropriately smooth functions on the rectangle H which interpolate the gridded data (3.4)-(3.6). It was found that the solution of this problem was a certain bicubic function with global smoothness $C^2(H)$. This problem was generalized to minimizing

$$(3.29)\quad \theta(f) = \int_a^b \int_c^d [f^{(m,n)}(x,y)]^2 dx\, dy, \quad m = 2p,\ n = 2q$$

in Ahlberg, Nilson and Walsh [1,2], whose solution involves certain higher-order polynomial splines. Since they are widely used, we give a short algebraic treatment here.

The points $\{x_i\}_0^{k+1}$ and $\{y_j\}_0^{\ell+1}$ define a partition of the intervals [a,b] and [c,d], respectively (cf. (3.5)). Suppose now that $x_{1-m} \le \ldots \le x_{-1} \le a < b \le x_{k+2} \le \ldots \le x_{k+m-1}$ and $y_{1-n} \le \ldots \le y_{-1} \le c < d \le y_{\ell+2} \le \ldots \le y_{\ell+n-1}$ are chosen arbitrarily. Let $\{N_i^m\}_{1-m}^k$ be the B-splines associated with the x-partition, and let the B-splines associated with the y-partition be denoted by $\{N_j^n(y)\}_{1-n}^{\ell}$. For a complete discussion of B-splines and their properties, see de Boor [36] in this volume (or [33]). Let

$$(3.30)\quad N_{ij}(x,y) = N_i^m(x) N_j^n(y), \quad i = 1-m,\ldots,k \text{ and } j = 1-n,\ldots,\ell.$$

The linear space

$$(3.31)\quad \mathcal{S} = \text{span}\, \{N_{ij}(x,y)\}_{i=1-m,\,j=1-n}^{k\ ,\ \ell}$$

is clearly of dimension $(k+m)(\ell+n)$. We may now construct an

element in (3.31) which interpolates to the gridded data.

Since there are only $(k+2)(\ell+2)$ data points on the grid (cf. (3.4)-(3.6)), it is clear that if we use s to interpolate, we have

$$(3.32)\quad (k+m)(\ell+n)-(k+2)(\ell+2) = (k+2)(n-2)+(\ell+2)(m-2)+(n-2)(m-2)$$

free parameters. Thus, to uniquely define a spline, one must add additional conditions. Recall that $m = 2p$ and $n = 2q$. Then we might add the extra conditions

$$(3.33)\quad \begin{aligned} s^{(\nu,0)}(x_0,y_j) &= s^{(\nu,0)}(x_{k+1},y_j) = 0, \quad \begin{matrix} j=0,1,\ldots,\ell+1 \\ \nu=p,\ldots,m-2 \end{matrix} \\ s^{(0,\mu)}(x_i,y_0) &= s^{(0,\mu)}(x_i,y_{\ell+1}) = 0, \quad \begin{matrix} i=0,1,\ldots,k+1 \\ \mu=q,\ldots,n-2 \end{matrix} \end{aligned}$$

and

$$(3.34)\quad \begin{aligned} s^{(\nu,\mu)}(x_0,y_0) &= s^{(\nu,\mu)}(x_0,y_{\ell+1}) = s^{(\nu,\mu)}(x_{k+1},y_0) \\ &= s^{(\nu,\mu)}(x_{k+1},y_{\ell+1}) = 0, \quad \begin{matrix} \nu=p,\ldots,m-1 \\ \mu=q,\ldots,n-1. \end{matrix} \end{aligned}$$

These are called the natural boundary conditions, and it can be shown that the system of equations

$$(3.35)\quad \sum_{i=1-m}^{k} \sum_{j=1-n}^{\ell} a_{ij} N_{ij}(x_\alpha,y_\beta) = F_{\alpha\beta}, \quad \begin{matrix} \alpha=0,1,\ldots,k+1 \\ \beta=0,1,\ldots,\ell+1 \end{matrix}$$

coupled with the conditions (3.33)-(3.34) provides a nonsingular system of equations for the coefficients $\{a_{ij}\}$. This system has convenient bandedness properties if the equations are arranged properly. The resulting spline is precisely the solution of the minimization problem (3.29). The boundary conditions (3.33)-(3.34) are the natural ones associated with the problem (3.29). However, it is also possible to specify lower-order derivative information along the boundary and also obtain a nonsingular system of equations. The resulting spline, called Type I, can also be shown to satisfy an appropriate minimization

problem. However, for data-fitting purposes, to use the interpolant with boundary derivative data one would first have to perform a first-stage approximation to find estimates for the required derivatives.

The best-known case of the above spline interpolation is the case $m = n = 4$, i.e., bicubic spline interpolation. In this case the surface constructed is a piecewise bicubic with global smoothness $C^2(H)$. The natural boundary conditions set second-derivative values to 0. Programs for computing natural bicubic interpolating splines can be found in the IMSL Library [117] in FORTRAN. FORTRAN programs for Type I bicubic splines can be found in Koelling and Whitten [121], where the required boundary information is assumed to be input. An ALGOL program for computing Type I bicubic splines in which boundary data are automatically computed by fitting one-dimensional splines appears in Späth [183].

Bicubic spline interpolation has been widely applied. For some references in the Geology literature, see eg. Anderson [7], Bhattacharyya [22], Holroyd and Bhattacharyya [115], Koelling and Whitten [121], and Whitten and Koelling [206].

Problem (3.29) has been widely generalized in the spline literature. Instead of minimizing ordinary derivatives, one may introduce general linear operators, and instead of dealing with point evaluation functionals, more general linear functionals may be permitted. To list some (but by no means all) papers dealing with such generalizations, we mention Arthur [8,9], Birkhoff, Schultz and Varga [29], de Boor [34], Delvos [65,66], Delvos and Schempp [68,69], Delvos and Schlosser [70], Fisher and Jerome [78,79], Haussmann [100], Haussmann and Munch [104], Munteanu [143,144], Nielson [148,150], Ritter [164,165], Sard [171,172], Schoenberg [176], Schultz [177,178], Späth [184,185], and Zavialov [209-212]. On L-shaped regions and other polygons

see Birkhoff [25] and Carlson and Hall [44-49].

We close this section by mentioning another direction of generalization which has led to a considerable development, the idea of spline blending. These methods are useful for construction of a surface which interpolates not only function values at isolated points but on the grid lines themselves; i.e.,

$$(3.36)\quad f(x,y_j) = F(x,y_j) \qquad a \le x \le b \quad \text{and} \quad j = 0,1,\ldots,\ell+1$$
$$f(x_i,y) = F(x_i,y) \qquad c \le y \le d \quad \text{and} \quad i = 0,1,\ldots,k+1.$$

To use such blending methods one must have F defined on the grid lines. Thus, the methods could be of value as second-stage processes. We do not have space to go into detail on spline-blended methods. We refer to the recent book of Barnhill and Riesenfeld [20] for a collection of papers on the subject and for further references. See also the papers of Gordon [84-87] and Gordon and Hall [88]. Recently, considerable effort has gone into showing that blending methods also arise as solutions of appropriate variational problems; see the papers of Delvos [65], Delvos and Kosters [66], and Delvos and Malinka [67].

4. Local interpolation methods

The interpolation methods discussed in section 3 were global in nature--that is, the value $f(x,y)$ of the constructed surface at any given point (x,y) in D depends on the values of all of the data points. This generally means that to compute a representation for f one has to solve a fairly large system of equations, and to evaluate $f(x,y)$ one generally has to carry out a considerable amount of arithmetic. In this section we shall consider local schemes where the surface depends only on nearby data points. Then the construction will usually lead to (a possibly large number) of <u>small</u> systems of equations, and moreover, the evaluation of the surface at a given point will

usually involve very little computation.

Many of the schemes mentioned in section 3 can be made local in nature by the following simple approach. Suppose that the domain D is partitioned into subdomains: $D = \bigcup_{i=1}^{d} D_i$. We then seek a surface in the form

$$(4.1) \quad f(x,y) = \{f_i(x,y), \quad (x,y) \in D_i, \quad i = 1,2,\ldots,d.$$

To construct each individual f_i, we suppose that $\tilde{D}_i$ are domains containing D_i, which contain only points which are "near" D_i. Then we use the data (and only the data) in $\tilde{D}_i$ to construct f_i. Usually, we can choose $\tilde{D}_i = D_i$. In most cases the most convenient choices for the subdomains D_i are triangles and rectangles. We discuss these two cases first.

4.1. Triangular subregions (scattered data). Suppose that we are given data at points $P_i = (x_i,y_i)$, $i = 1,2,\ldots,N$ scattered throughout the plane, and let D be the convex hull of these points. It is more or less clear that by drawing lines from point to point we can construct a set of triangles with vertices at the P_i which partition D. It is also clear that given any set of points, this triangularization of D is not usually uniquely defined (see Figure 2 below for two different triangularizations of the same region). Moreover, as the figure shows, some triangularizations are superior to others in the sense that they exhibit fewer of the less desirable long thin triangles.

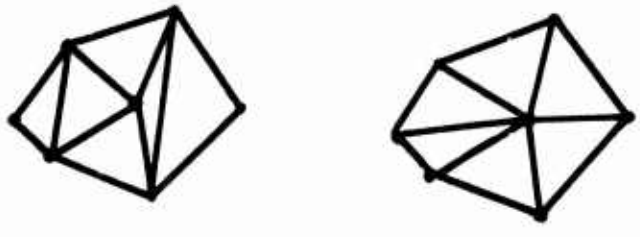


Figure 2. Triangularization

The design of an algorithm to divide a region into acceptable triangles with vertices at prescribed points is not as easy as it sounds. Two algorithms in the literature which are designed to give good triangularizations can be found in Cavendish [50] and in Lawson [128].

The simplest approach to defining a local interpolating surface is to construct $f_i(x,y)$ to be of the form $a_1 + a_2x + a_3y$ in each triangle. The data at the three corners of the triangle determine the coefficients for that piece of f (the corresponding system will be nonsingular provided the triangle is nondegenerate). This procedure produces a piecewise linear surface which, in fact, will be globally continuous. This last property follows from the fact that along the sides of the triangle the functions reduce to straight lines joining the vertices. This method has been used by several authors for data fitting, e.g., Lawson [128] and Whitten [206]. For some contouring routines based on this local interpolation scheme, see section 8.

If we desire to interpolate several sets of data defined on the same triangularization, it may be more convenient to compute Lagrangian functions rather than to compute the surface in each triangle separately. In particular, it is clear that we can construct functions $\{\phi_j(x,y)\}_1^N$ with the property

$$(4.2)\quad \phi_j(x_i,y_i) = \delta_{ij}, \quad i,j = 1,2,\ldots,N.$$

These functions can be constructed as pyramids in such a way that the function ϕ_j has support only on the triangles surrounding the point (x_j,y_j) (see Figure 3). In terms of these Lagrangian functions, the interpolating surface is given by

$$(4.3)\quad f(x,y) = \sum_{j=1}^{N} F_j\phi_j(x,y).$$

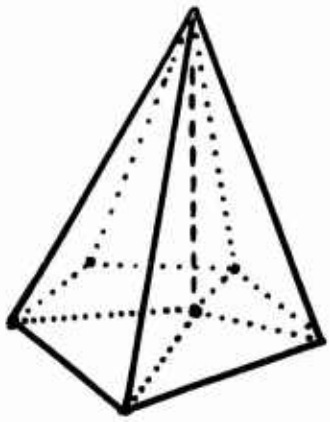


Figure 3. A Lagrange Element

The Lagrangian approach to local interpolation is very reminiscent of the finite element method in which the solution of an operator equation is sought in the form of a linear combination of a set of functions (called elements) with the property (4.2). (See e.g., Prenter [157], Schultz [179], or Strang and Fix [188].) There is no need to restrict the elements to be piecewise linear functions--we may use higher-order polynomials, rational functions, or even more complicated functions. In fact, if we are careful in the construction, we may be able to construct elements with small support but higher global smoothness.

There are a great many papers in the finite-element literature concerned with defining convenient smooth elements (Lagrangian functions with small support). To mention a few, see Barnhill, Birkhoff, and Gordon [16], Barnhill and Gregory [17, 18], Barnhill and Mansfield [19], Birkhoff and Mansfield [28], Bramble and Zlamal [39], Goel [83], Hall [94], Mitchell [141], Mitchell and Phillips [142], Nicolaidis [146,147], Zenisek [213], Zienkowicz [214], and Zlamal [215-217]. The books on finite elements of Aziz [13], de Boor [35], Strang and Fix [188], and Whiteman [198] should also be consulted.

The construction of elements with higher-order smoothness becomes increasingly difficult. For example, it is shown in Mansfield [137] that to get an element with support on the triangles surrounding P_j and with global continuity $C^1(D)$, it is necessary to use polynomials of degree 5 at least. (Matters are somewhat simpler on regular triangularizations, see subsection 4.2 below.)

We close this subsection by mentioning that it is also possible to perform interpolation using elements based on triangles to data which also involves derivatives, or in analogy with the blending methods, to data which includes function values along the edges of the triangles. (See e.g., Barnhill, Birkhoff, and Gordon [16], or Barnhill and Gregory [17,18].) These methods are not directly applicable to the scattered data Problem 1.1, but may be useful as second-stage methods.

4.2. Regular triangularizations. When the data is distributed such that the region can be triangulated into a set of congruent triangles, then it is extremely advantageous to use the Lagrange approach. In particular, in this case we can find an element ϕ with value 1 at (0,0) such that all other elements are translates of ϕ. In this case, f takes the form

$$(4.4) \quad f(x,y) = \sum_{j=1}^{N} F_j \phi((x,y) - (x_j,y_j)).$$

We illustrate this with a couple of examples. Suppose that we are given data at points chosen from the collection

$$(4.5) \quad \Omega_1 = \{(i,j)\}_{i,j\in Z} \cup \{(i+\tfrac{1}{2}, j+\tfrac{1}{2})\}_{i,j\in Z}, \quad Z = \{\text{integers}\}.$$

These points lie on the corners of a triangular grid as shown in Figure 4.

It is shown in Zwart [218,p.673] that there exists a function $\phi \in C^1(R^2)$ which is 1 at the origin and 0 at all other points in Ω, and has support on the shaded region in Figure 4.

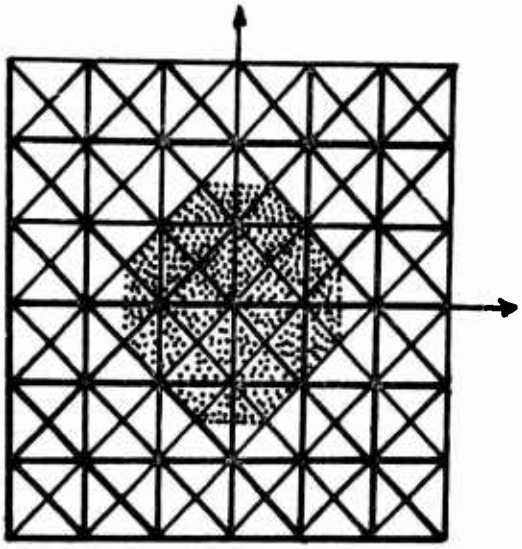


Figure 4. A Regular Triangularization

This function is constructed as a piecewise quadratic polynomial. A similar element has been constructed by Powell [156] (the figure on page 267 of [156] should be rotated 45° to see this).

To give another example, suppose that we consider the set of points Ω_2 which lie at the vertices of the grid defined by equilateral triangles shown in Figure 5.

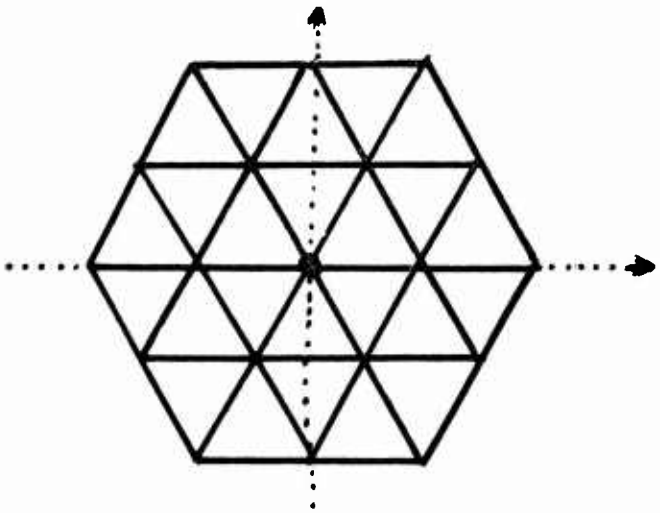


Figure 5. Another Regular Triangularization

It is shown in Fredrickson [81] that there exists a function ϕ which has value 1 at the origin and value 0 at all other points in Ω_2. The function ϕ is in $C^2(R^2)$, consists of piecewise quartics, and has support in the region shown in Figure 5. Fredrickson also constructs a piecewise cubic element with the same support but which is only $C^1(R^2)$. For right triangles see Carlson and Hall [44].

4.3. Rectangular subregions. In this section we suppose that we have data given at points lying on a rectangular grid as in (3.4)-(3.6), and consider local interpolation methods. The simplest approach here (cf. the triangularization case) is to construct a separate bilinear function $f(x,y) = a_1 + a_2x + a_3y + a_4xy$ in each subrectangle, $H_{ij} = [x_i,x_{i+1}] \times [y_j,y_{j+1}]$, using the four corner values to determine the coefficients. Since the bilinear patches reduce to linear functions on the grid lines, the global surface is $C(R)$.

Several authors have considered constructing functions on each of the H_{ij} using higher-order polynomials. This requires additional information in addition to the four corner values. For example, if one seeks a bicubic

$$(4.6) \quad f(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j,$$

there are 16 coefficients to determine. These could be determined by the four corner values, plus the values of f_x, f_y, and f_{xy} at each corner. To determine these, one must perform some first-stage process. For some approaches to this, see Akima [5], Hessing, et al [114], and Shu, et al [181]. A FORTRAN program for Akima's method can be found in [6]. Nonpolynomial patches have also been considered; e.g., see Birkhoff and Garabedian [27].

The Lagrange (finite element) approach can also be used in

the case of rectangular gridded data. In particular, if we can construct a function satisfying (4.2) with local support, then the surface f given by (4.3) will interpolate and the method will be local in character. As before, the Lagrange approach is especially convenient if the grid is regular, i.e., if all subrectangles H_{ij} are congruent. To illustrate this, suppose that the H_{ij} are actually the unit squares; i.e., the data points lie in the set

(4.7) $\Omega_3 = \{(i,j)\}$ $i,j \in Z$, $Z = \{\text{integers}\}$.

To get a quadratic C^1 element, we may simply rotate the element of Zwart [218] considered in the last section by 45 degrees (cf. Figure 4), or we may take the element of Powell [156].

4.4. Parametric representations. The methods discussed in the last section is concerned with data given on a rectangular grid. By using parametric representations, it is possible to construct similar local interpolating surfaces for data given at the corners of any partition of D consisting of quadrilaterals. In this section we briefly describe how this might proceed.

Suppose Q is a particular quadrilateral subregion of D of interest. In addition, suppose that $x(s,t)$, $y(s,t)$, and $z(s,t)$ are functions defined on the unit square $U = [0,1] \times [0,1]$ with the properties that as (s,t) runs over the boundary of U, $(x(s,t),y(s,t))$ runs over the boundary of the quadrilateral; the four corners of U correspond to the four corners of Q; and $z(s,t)$ takes on the desired data values at the four corners of U. In this case, the triple $(x(s,t),y(s,t),z(s,t))$ provides a parametric representation of a piece of surface defined over Q interpolating the data.

The problem of constructing parametric representations of interpolating functions has been considered in a number of papers. Several papers on these methods and a host of references can be found in the book of Barnhill and Riesenfeld [20]; see

also the survey paper of Shu et al [181]. Such surfaces are sometimes called Coon's surfaces, cf. Coons [59], and are of considerable interest in the field of computer-aided geometric design. To mention just a few of the actual papers, see Ahuja and Coons [4], Earnshaw and Youille [74], Ferguson [77], Hayes [107], Hosaka [116], and Mangeron [132].

There also has been some effort directed towards constructing elements (Lagrange functions) associated with other less regular subsets of the plane. We mention, for example, the work of Ciarlet and Raviart [55], Wachspress [194,195], and Zlamal [217] in which elements are constructed for domains involving curved edges.

4.5. Local Shepard methods. It is possible to modify the method discussed in subsection 3.3 to make it local. For example, following Shepard [180], suppose we fix $0 < R$ and define

$$(4.8)\quad \psi(r) = \begin{cases} 1/r & 0 < r \le \frac{R}{3}, \\ \frac{27}{4R}\left(\frac{r}{R} - 1\right)^2, & R/3 < r \le R, \\ 0, & R < r. \end{cases}$$

This function is continuously differentiable and vanishes identically for $r < R$. Now with r_i as in (3.8), we define

$$(4.9)\quad f(x,y) = \begin{cases} \dfrac{\sum_{i=1}^{N} F_i [\psi(r_i)]^{\mu}}{\sum_{i=1}^{N} [\psi(r_i)]^{\mu}}, & \text{when } r_i \neq 0, \text{ all } i \\ F_i, & \text{when } r_i = 0. \end{cases}$$

Formula (4.9) is defined at all (x,y) in the plane R^2. By definition it interpolates the values F_i at the data points (x_i,y_i), $i = 1,2,\ldots,N$. The values at non-data points are obtained as weighted averages of the data values F_i, but

only those which lie at points within a distance of R of (x,y). Thus, the formula is local.

To use this method in practice it is necessary to choose a reasonable value for R. The aim is to find R so that for every (x,y) a reasonable number of data points will fall in the disk centered at (x,y) of radius R. It would also be possible to let R depend on (x,y), i.e., to use different values of R in different subregions of D.

5. Global approximation

As mentioned in the introduction, frequently the data does not warrant constructing an interpolating function (e.g., because of errors). In such cases it may be preferable to construct a surface which only approximates the data. In this section we discuss some global approximation methods.

5.1. Polynomial least squares. The general theory of discrete least-squares fitting is very well known. To briefly review, suppose that $\{\phi_j\}_1^n$ are n given functions on D. Define

$$\Phi(a) = \sum_{i=1}^{N} \left| \sum_{j=1}^{n} a_j \phi_j(x_i,y_i) - F_i \right|^2, \tag{5.1}$$

where $a = (a_1,\ldots,a_n)^T$ is any vector in R^n. Then the problem is to find a^* such that

$$\Phi(a^*) = \min_a \Phi(a). \tag{5.2}$$

The corresponding function

$$f(x,y) = \sum_{j=1}^{n} a_j^* \phi_j(x,y) \tag{5.3}$$

is called the discrete least-squares approximation of the data $\{F_i\}_1^N$. Usually one takes n considerably smaller than N. In this section we briefly discuss least squares using polynomials. Before doing so, however, we make a few general remarks about

solving the general least-squares problem.

There are several approaches to solving (5.2). Perhaps the neatest is the case where the $\{\phi_j\}_1^n$ are orthonormal with respect to the inner-product

$$(5.4) \quad (\phi,\psi) = \sum_{i=1}^{N} \phi(x_i,y_i)\psi(x_i,y_i).$$

Then the solution of (5.2) can be written down explicitly as

$$(5.5) \quad f(x,y) = \sum_{j=1}^{n} F_j \phi_j(x,y).$$

A second very well-known approach to solving (5.2) is via the normal equations

$$(5.6) \quad A^*A\,a = A^*F,$$

where $F = (F_1,\ldots,F_N)^T$ is the vector of data values, and where

$$(5.7) \quad A = (\phi_j(x_i,y_i))_{j=1,i=1}^{n,N}.$$

In some cases the normal equations are a perfectly acceptable way to compute least-squares approximation, but in other cases the system (5.6) may be ill-conditioned (or even singular--cf. the following subsection for spline least squares). This approach is also not convenient should side conditions be desired (e.g., by imposing actual interpolation at some of the values). For more on the normal equations, see any book on Numerical Analysis.

A more modern method of solving least-squares problems is to use general matrix methods. Specifically, consider the observation equations

$$(5.8) \quad A\,a = F.$$

It can be shown that by applying a series of matrix transformations to this system, one can obtain a set of equations giving the vector a^*. For a complete description of methods of this

type see Lawson and Hanson [129] or Stewart [187]. Matrix methods are quite amenable to the adding of side conditions and can also be designed to take account of rank-deficiency of the matrix A (which corresponds to the case of singular normal equations).

Polynomial discrete least-squares fitting has been widely used for fitting surfaces to data, both scattered and regular. Several authors have developed algorithms for polynomial discrete least-squares fitting of scattered data by constructing orthonormal polynomials (e.g. by Gram-Schmidt orthonormalization). See, for example, Cadwell and Williams [42], Crain and Bhattacharyya [61], and Whitten [201,202]. The latter contains a FORTRAN program.

When the data are more regularly distributed, polynomial least-squares fitting can often be simplified. For example, if the data lie on a grid as in (3.4)-(3.6), then the desired orthogonal polynomials are simply products of the one-dimensional orthogonal polynomials associated with the one-dimensional inner products corresponding to $\{x_i\}_0^{k+1}$ and $\{y_j\}_0^{\ell+1}$ respectively; e.g., see Cadwell [41] or Clenshaw and Hayes [56], as well as the survey papers of Hayes [105,108,109].

There are also special methods for handling data which are not on a grid but instead lie on parallel straight lines. For example, Clenshaw and Hayes [56] have developed methods using expansions in terms of Tchebycheff polynomials (although the method actually only produces an approximation to the least-squares fit rather than the actual minimum).

Polynomial least squares can also be interpreted as multi-dimensional regression as practiced by statisticians, e. ., see Effroymson [75]. For example, if we are trying to fit a function in the form

$$f(x,y) = \sum_{i=0}^{dx} \sum_{j=0}^{dy} a_{ij} x^i y^j,$$

then by defining new variables by

$$z_{\nu(dy+1)+\mu} = x^\nu y^\mu, \qquad \begin{matrix} \nu = 0,1,\ldots,dx \\ \mu = 0,1,\ldots,dy \end{matrix}$$

we can write

$$f(x,y) = \sum_{i=0}^{d} b_i z_i, \qquad d = dxdy + dx + dy,$$

and the problem becomes one of fitting a <u>linear</u> function in several variables.

We close this section by observing that in some cases it may be desirable to consider <u>weighted</u> least squares. In particular, if we have positive weights $w_i > 0$, $i = 1,2,\ldots,N$, then we may replace Φ in (5.1) by

$$I_w(a) = \sum_{i=1}^{N} w_i \left| \sum_{j=1}^{n} a_j \phi_j(x_i,y_i) - F_i \right|^2 .$$

It is interesting to note that the interpolation formula of Shepard discussed in section 3.3 can be interpreted in terms of weighted least-squares fitting. In particular, fix (x,y) in D, and let $r_i(x,y)$ be the distance from (x,y) to the point (x_i,y_i) as before. Now set $w_i = r_i^{-\mu}$, and consider least-squares approximation by a constant c, using these weights. Then one easily computes that the least-squares choice of c is

$$c = \frac{\sum_1^N w_i F_i}{\sum_1^N w_i} = \frac{\sum_1^N F_i r_i^{-\mu}}{\sum_1^N r_i^{-\mu}} .$$

This approach was adopted by Pelto, Elkins and Boyd [152] (as pointed out to me by Chuck Duris).

5.2. Discrete least-squares fitting by splines. As outlined in the previous subsection, discrete least squares can be carried out with any finite set of functions. It is not surprising that a number of authors have tried using tensor product splines. See, e.g., Halliday, Wall, and Joyner [96], Hayes and Halliday [110], Jordan [119], Hanson, Radbill, and Lawson [97], and Whiten [199]. Hayes and Halliday have developed both ALGOL and FORTRAN programs. It is, on the other hand, perhaps somewhat surprising that least-squares fitting with splines can be somewhat problematical. We briefly discuss the method.

Suppose that $H = [a,b] \times [c,d]$ is a rectangle containing the domain D of interest. Let $\{x_i\}_0^{k+1}$ and $\{y_j\}_0^{\ell+1}$ be partitions of $[a,b]$ and $[c,d]$, respectively, and let $\{N_{ij}\}_{1-m,1-n}^{k+1\ \ell+1}$ be the tensor product B-splines discussed in section 3.5. We consider discrete least-squares fitting using these $(k+m)(\ell+n)$ B-splines.

To explain how difficulties can arise with spline least-square fitting, we observe that it is quite easy for the matrix A in the observational equations (5.8) to be rank-deficient. On a trivial level this can happen if for some B-spline N_{ij}, none of the data points lies in its support. This deficiency can, of course, be easily removed by dropping this particular B-spline from the set being used to approximate. But rank deficiency can also occur in more subtle ways because of the local support properties of the functions. This problem can be overcome with properly designed algorithms. See Hayes and Halliday [110] for a careful discussion of spline least-squares fitting. Lawson and Hanson [129] include a general discussion of how to handle rank deficient least-squares problems.

If we operate in terms of the normal equations, then it may well occur that the normal equations are in fact singular. This is again due to the local property of the B-splines com-

bined with the discrete inner-product. Even when it is not singular, the set of normal equations can be ill-conditioned (even though it is a relatively sparse matrix with a kind of repeated band-structure).

Discrete least squares can also be carried out with various finite dimensional linear spaces of blended functions. For an extensive study of such methods, see the dissertation of Doty [71].

5.3. Discrete ℓ_1 and ℓ_∞ approximation. Instead of performing discrete least squares, we may consider the following discrete approximation problem: Given functions $\{\phi_j\}_1^n$ defined on D, we seek a^* so that

$$(5.9)\quad \Phi(a) = \sum_{i=1}^{N} \left| \sum_{j=1}^{n} a_j \phi_j(x_i, y_i) - F_i \right|$$

is minimized. Alternatively, we may minimize

$$(5.10)\quad \Phi(a) = \max_{1 \le i \le N} \left| \sum_{j=1}^{n} a_j \phi_j(x_i, y_i) - F_i \right|.$$

These are the usual ℓ_1 and ℓ_∞ best approximation problems. Both of these problems can easily be reformulated as linear programming problems for the determinations of the optimal a^* (cf. Rabinowicz [160,161] or Rosen [167]). Reasonable choices for the $\{\phi_j\}$ would be low-degree polynomials if D is small, or possibly spline functions.

Discrete approximation methods of this type have had relatively little exposure in the literature. For some results using tensor product splines in the ℓ_∞ problem, see Rosen. The optimal a^* was obtained there by using the standard simplex method on the associated dual linear programming problem.

The ℓ_∞ problem can also be solved by using Remez-type algorithms. For an algorithm which performs generalized

rational approximation (and thus can also be used for polynomial approximations) see Kaufman and Taylor [120]. Theoretical considerations for Tchebycheff approximation in several variables can be found in Collatz [58] or Weinstein [196], for example.

5.4. Spline smoothing (scattered data). In this section we consider some minimization problems similar to those discussed in section 3.4, but where the class of admissible functions is not required to interpolate and where the functional to be minimized includes a term measuring how close the function comes to fitting the data. To be more specific, suppose X is a linear space of "smooth" functions and that θ is a functional on X which measures the smoothness of an element in X. Suppose in addition that E is a functional defined on X which measures how well a function fits the data. Then the spline-smoothing problem is the following:

(5.11) Find $s \in X$ such that $\rho(s) = \inf_{u \in X} \rho(u)$,

where

(5.12) $\rho(f) = \theta(f) + E(f)$.

The abstract theory of spline smoothing has been well developed; see, e.g., the book of Laurent [127] and references therein. To illustrate the ideas, we briefly discuss a couple of examples. We suppose as in section 3.4 that X is a semi-Hilbert space and that θ is a seminorm on X with $N = \{f \in X: \theta(f) = 0\}$. We also suppose that X is actually a function space defined on a domain D, and that the point evaluators at $\{(x_i,y_i)\}_1^N$ are bounded linear functionals on X. We define

(5.13) $E(f) = p \sum_{i=1}^{N} [f(x_i,y_i) - F_i]^2$,

where p is a fixed positive constant. Then it can be shown

(cf. Duchon [72,73]) that the solution of Problem (5.11) is a spline which can be written in the form (3.20), where now the coefficients are determined from the linear system

$$\sum_{i=1}^{N} K((x_j,y_j);(x_i,y_i))a_i + \sum_{i=1}^{d} b_i p_i(x_j,y_j) + a_j/p = F_j,$$

(5.14) $$j = 1,2,\ldots,N,$$

$$\sum_{i=1}^{d} a_i p_k(x_i,y_i) = 0, \qquad k = 1,2,\ldots,d.$$

As in section 3.4, the application of this method depends on constructing a reproducing kernel K. If θ is chosen as in (3.22), Atteia [10-12] and Thomann [192,193] considered spline smoothing for spaces of smooth functions on the rectangle and on the disc (the latter even contains ALGOL programs). Duchon [72,73] considers similar problems defined on $D = R^2$.

A similar spline-smoothing problem has also been considered by Pivorarova [154], where θ is taken to be

$$(5.15) \quad \theta(f) = \iint [D_x^2 f]^2 + [D_y^2 f]^2.$$

See also Kubik [123].

5.5. Smoothing splines (gridded data). In section 3.5 we considered several minimization problems whose solutions led to interpolating polynomial splines (and generalizations). In conjunction with the development of interpolating splines for gridded data, there was a concurrent development of smoothing splines. For example, instead of minimizing the integral θ in (3.29) over appropriate smooth interpolating functions, we may minimize instead $\rho(f) = \theta(f) + pE(f)$, where E is given by

$$(5.16) \quad E(f) = \sum_{i=0}^{k+1} \sum_{j=0}^{\ell+1} [f(x_i,y_j) - F_{ij}]^2.$$

For results in this direction, see e.g. Nielson [149,150]. For

Θ given by (3.29), the smoothing splines are again polynomial splines. Again, more general linear differential operators and more general linear functionals can be considered.

5.6. Continuous least squares. The method of continuous least squares is not directly suited to fitting surfaces to discrete data, but it can be of use as a second-stage process, so we briefly review it. We suppose now that F is a function defined on D which we wish to approximate, and that $\{\phi_j\}_1^n$ are given functions on D. We define

$$(5.17)\quad \langle f,g\rangle = \iint_D f(x,y)\,g(x,y)\,dxdy, \quad \|f\|^2 = \langle f,f\rangle$$

and

$$(5.18)\quad \Phi(a) = \Big\| \sum_{j=1}^{n} a_j\phi_j - F\Big\|^2.$$

The problem is to find a^* to minimize $\Phi(a)$. The solution is given by solving the normal equations

$$(5.19)\quad Aa = r,$$

where

$$A = (\langle \phi_i,\phi_j\rangle)_{i,j=1}^{n} \quad \text{and} \quad r = [\langle \phi_1,F\rangle,\ldots,\langle \phi_n,F\rangle]^T.$$

For reasonably nice approximating functions it is often possible to compute the normal matrix exactly. In practice, the difficulty lies in evaluating the right-hand sides. Generally a quadrature formula is required for this. One advantage of the method would be that if several data-fitting problems are to be solved using the same set of approximating functions, one can do the work of inverting the normal matrix just once and re-use the result as many times as desired.

Reasonable choices for the approximating functions include polynomials, or better yet, tensor product B-splines as in (3.30). Here the singularity problems do not crop up for the splines because we are integrating instead of summing over

finitely many points. The normal matrix in this case has a kind of repeated band structure. The entries can be computed exactly, e.g., by Gaussian quadrature (cf. de Boor, Lyche and Schumaker [38]). Uniform best approximation by tensor products of splines has also been considered, e.g., see Sommer [182].

6. Local approximation methods

As pointed out at the beginning of section 4, there are many advantages which accrue if one uses local methods rather than global ones. In this section we discuss some local approximation schemes.

6.1. Patch methods. As in the case of interpolation, the simplest approach to obtaining local approximation methods is to partition the domain and to define a surface (patch) on each subdomain separately. In particular, suppose that $D = \cup\{D_i\}_1^d$, where D_i are disjoint subsets of D. Then we may seek f in the form

$$(6.1) \quad f(x,y) = \{f_i(x,y), \ (x,y) \in D_i, \quad i = 1,2,\ldots,d.$$

To construct the patch $f_i(x,y)$, we might use the data available in the subregion D_i. In certain cases, however, it may well occur that no data at all are available in the set D_i. In this case we may choose a somewhat larger set $\tilde{D}_i$ of points "near" D_i, and use the data in $\tilde{D}_i$ to construct f_i. For any given method, it should be possible to make the choice of $\tilde{D}_i$ adaptive so that the size of $\tilde{D}_i$ is kept as small as possible consistent with the amount of data desired for the construction of f_i.

The patch method outlined above can be used with any of the approximation methods discussed in section 5. For example, one might choose to use polynomials (of low order), and to do discrete least-squares approximation. Or, one might use ℓ_1 or ℓ_∞ approximation or some other convenient space (e.g. splines)

instead of polynomials. The main point is to keep the size of each individual patch problem (and thus the size of the corresponding system of equations) small. We may have to solve a lot of systems of equations, but each will be small and fairly well-conditioned.

To illustrate how the adaptive feature might be implemented, suppose that the domain D of interest has been enclosed in a rectangle H and that a partition of H is defined by $H = \cup\{H_{ij}\}_{i=0,j=0}^{k,\ell}$, with $H_{ij} = [x_i,x_{i+1}] \times [y_j,y_{j+1}]$ for some

$$(6.2)\quad a = x_0 < x_1 < \ldots < x_{k+1} = b,\ c = y_0 < y_1 < \ldots < y_{\ell+1} = d.$$

Now suppose that we want to do discrete least-squares fitting using a patch of the form $f_{ij}(x,y) = a + bx + cy$ on H_{ij}. In this case it would be reasonable to require that at least 3 pieces of data should be used to construct f_{ij}. If H_{ij} does not contain 3 pieces of data, we expand H_{ij} to $\tilde{H}_{ij}$ by adding all bordering rectangles. If this does not contain 3 pieces, we again add all bordering rectangles, etc. We then compute the discrete least-squares polynomial using the data in $\tilde{H}_{ij}$, but then we use the resulting function only in H_{ij}. The process may be repeated to define each required patch. This kind of adaptive algorithm is very easy to program.

In using patch methods to get local interpolation methods, we concentrated on methods using data at corners of triangles or rectangles, and by choosing appropriate forms for the patches, it was possible to get the individual patches to match together to give a continuous global surface (or with more sophisticated patches, even $C^1(D)$ or higher). Here, however, where the individual patches are determined by approximation, it is not very likely that the patches will match up, and the global surface will generally not even be continuous. For most applications, this is a serious drawback. However, as we shall see in

section 7, patch approximation methods can still be very useful as first-stage methods.

6.2. Direct local methods. In this section we discuss some local methods in which the approximating surface is constructed directly from the data without solving any systems of equations. It will be convenient to pose a more general problem than previously considered.

Let $\mathcal{F}$ be a linear space of functions defined on D, and suppose that $\{\lambda_i\}_1^N$ are linear functionals defined on $\mathcal{F}$. Let $\{\phi_i\}_1^N$ be a prescribed set of functions defined on D. Then we are interested in approximation schemes of the following form:

$$(6.3)\quad QF(x,y) = \sum_{i=1}^{N} \lambda_i F \phi_i(x,y).$$

We can think of this as a surface-fitting problem where the data are given by $F_i = \lambda_i F$, $i = 1,2,\ldots,N$. Given the data, we can write the approximation down immediately.

We also observe that if the ϕ_i have support on small subsets of D, and if each λ_i also has support on the same set, then the formula (6.3) is local. For example, if we take λ_i to be point evaluation at the point (x_i,y_i) and $\phi_i(x,y)$ to be a function with support in a neighborhood of (x_i,y_i), then the approximation formula simply becomes

$$(6.4)\quad QF(x,y) = \sum_{i=1}^{N} F_i \phi_i(x,y).$$

This is very reminiscent of the Lagrange form of interpolation (cf. (4.3)), but unless the ϕ_i are taken to satisfy (4.2), QF will not in fact be an interpolant. For this reason, formulae of the form (6.4) (or more generally (6.3)) are sometimes referred to as quasi-interpolants. Local quasi-interpolants of the form (6.3) can be constructed simply by defining the

functions $\{\phi_i\}_1^N$ with local supports. If each of these is continuous (or smooth), then QF will also be.

Although a host of quasi-interpolants can be constructed as outlined above, considerable care must be exercised in order to get methods which give good accuracy (when the original function F is smooth). As observed earlier, this is directly related to making the method exact for polynomials, i.e., such that QP = P for all P in some class of polynomials.

To construct methods of the form (6.3) which apply to scattered data, it is necessary to construct appropriate $\{\phi_i\}_1^N$. While a host of methods can be constructed this way, it is not so easy to choose the ϕ_i to make the method exact for polynomials (which, as we remarked earlier, is directly related to how well the method will approximate smooth functions F). To get methods which do have a reasonable degree of exactness (and a correspondingly good error bound for smooth functions), it is easier to first choose the $\{\phi_i\}_1^N$, and then try to find suitable $\{\lambda_i\}_1^N$. While this generally rules out using point evaluators at scattered data, it is possible to construct methods based on point evaluators at appropriate points, and such methods can be useful as <u>second-stage</u> approximations.

To illustrate these ideas, we consider construction of local spline approximation methods following the general treatment in Lyche and Schumaker [131]. Suppose D is enclosed in a rectangle H, and that H is partitioned into subrectangles by a grid as in (6.2). Suppose that $\{N_{ij}\}_{i=1-m,\,j=1-n}^{k,\,\ell}$ are the tensor product B-splines associated with this partition (cf. (3.30)). We are now interested in approximation schemes of the form

$$(6.5)\quad QF(x,y) = \sum_{i=1-m}^{k} \sum_{j=1-n}^{\ell} \lambda_{ij} F N_{ij}(x,y).$$

In particular, we are going to consider the question of

constructing formulae of this type which are exact for the class of polynomials $\mathcal{P}_{v,u}$, with some fixed $1 \le v \le m$ and $1 \le u \le n$. This problem has a very simple algebraic solution if we decide to construct each λ_{ij} in the form

$$(6.6)\quad \lambda_{ij} = \sum_{\nu=1}^{v} \sum_{\mu=1}^{u} \alpha_{ij\nu\mu}\, \lambda^{x}_{ij\nu}\, \lambda^{y}_{ij\mu},$$

where the $\{\lambda^{x}_{ij\nu}\}_{\nu=1}^{v}$ and $\{\lambda^{y}_{ij\mu}\}_{\mu=1}^{u}$ are linear functionals which apply to functions of x and y alone, respectively. It can be shown (cf. [131]) that given any $\{\lambda^{x}_{ij\nu}\}$ and $\{\lambda^{y}_{ij\mu}\}$ satisfying mild independence assumptions, there exist coefficients $\{\alpha_{ij\nu\mu}\}$ such that the formula (6.5) will be exact for $\mathcal{P}_{vu}$. In fact, these coefficients can easily be explicitly computed.

To give one example, suppose

$$(6.7)\quad \begin{cases} \xi_i = \dfrac{(x_{i+1}+\ldots+x_{i+m-1})}{(m-1)}, & i = 1\text{-}m,\ldots,k \\ \eta_j = \dfrac{(y_{j+1}+\ldots+y_{j+\ell-1})}{(n-1)}, & j = 1\text{-}m,\ldots,\ell. \end{cases}$$

Then we obtain

$$(6.8)\quad QF(x,y) = \sum_{i=1-m}^{k} \sum_{j=1-n}^{\ell} F(\xi_i,\eta_j) N_{ij}(x,y),$$

a formula which exactly reproduces the bilinear polynomials $\mathcal{P}_{1,1}$. This is the multidimensional (tensor product) version of the Variation Diminishing method of Marsden and Schoenberg; it was studied in some detail in Munteanu and Schumaker [145]. This formula is closely related to the Bezier-type surfaces constructed in Riesenfeld [163] (see also Gordon and Riesenfeld [89]).

We should observe that the way formula (6.5) now stands, it may involve information on F which is taken from data outside of the domain D. This situation can be rectified as follows:

Let

(6.9) $\Omega = \{(i,j):$ support $\lambda_{ij} \cap D$ not empty$\}$.

Then it can be shown [131] that the method

$$(6.10)\quad QF(x,y) = \sum\sum_{(i,j)\in\Omega} \lambda_{ij} FN_{ij}(x,y)$$

remains exact as long as all functions are restricted to D.

To get higher-order methods, depending only on point evaluations, we proceed as follows. Choose

$$(6.11)\quad \begin{aligned} x_i < \tau^x_{ij\nu} < x_{i+m}, \quad & \nu = 1,2,\ldots,v \\ y_j < \tau^y_{ij\mu} < y_{j+n}, \quad & \mu = 1,2,\ldots,u, \end{aligned}$$

for $i = 1\text{-}m,\ldots,k$ and $j = 1\text{-}n,\ldots,\ell$. Then if we take $\lambda^x_{ij\nu}$ to be point evaluation at $\tau^x_{ij\nu}$ and $\lambda^y_{ij\mu}$ to be point evaluation at $\tau^y_{ij\mu}$, the coefficients in (6.6) are easily computed. Hints on where the τ's should be placed within the support of the B-splines are given by the error analysis in [131].

We close this section with some historical remarks on the development of local approximation schemes in two dimensions. Early papers include Babuska [14], de Boor and Fix [37], and Fix and Strang [80]. For some methods involving triangular partitions, see Fredrickson [82]. Quasi-interpolants were constructed in de Boor and Fix [37] using point evaluation data, but including derivatives. We have followed Lyche and Schumaker [131] where general linear functionals are considered, and where in particular, methods can be constructed using only point evaluation of the function. (Local integrals etc. would also be possible.) The papers [37] and [131] both contain extensive error bound analyses. It is striking that these local spline approximation methods give optimal order error bounds for smooth functions.

7. Two-stage processes

Many of the methods we have discussed in this paper are only applicable when the data are regularly spaced (and in fact, many surface-fitting methods require specification of derivative data as well as function values). Such methods cannot be applied directly to the scattered data-fitting Problem 1.1. On the other hand, some of the most convenient local interpolating and local approximating methods which do work for scattered data produce surfaces which are not globally smooth (or even continuous). Thus, it seems natural to consider the possibility of constructing two-stage processes in which the first stage uses the scattered data to construct an approximation g, while the second stage uses g to generate data for constructing a surface f (with desirable properties, such as smoothness).

Since it is quite clear how various methods discussed in the earlier sections might be put together to yield two-stage processes, it will suffice to mention just a couple of examples here.

7.1. Interpolation/interpolation. Suppose that we want to construct a piecewise bicubic surface based on data given on a rectangular grid as in (3.4)-(3.6). In each subrectangle H_{ij} the 16 coefficients of the bicubic f (cf. (4.6)) would be determined by the values of f, f_x, f_y, and f_{xy} at each of the four corners. Now since our original data-fitting problem only specifies the values of the function at the grid points, local interpolation cannot be carried out directly. However, we can use the data to provide estimates for the values of f_x, f_y, and f_{xy} at the grid points (i.e., we construct g interpolating the data); then we can use local bicubic interpolation as a second stage. The reader will have no difficulty in imagining ways to produce estimates for these quantities. For some methods which appear in the literature, see the papers of Akima

[5,6], Koelling and Whitten [121], and Späth [183].

7.2. Approximation/interpolation. Instead of making the first-stage process interpolation as in section 7.1, it would also be possible to use an approximating process. For example, one might use least-squares polynomial approximation to construct a patch surface and then use some convenient interpolation process as a second stage. For an example of this type, see Hessing et al [114].

7.3. Approximation/approximation. This combination is particularly convenient if we are not concerned about getting an interpolating function. Both stages can be made local. To give an example, recently I have constructed an algorithm for fitting surfaces to scattered data in which the first stage consists of polynomial least-squares patch approximation (with adaptive choice of data--see section 6), and where the second stage consists of direct local tensor product spline approximation. Both stages are local, and the final surface is a tensor product spline. Since the second stage is a direct method, it is very cheap to apply. Experiments with real-life data (e.g. from heart potentials, potential fields, and geological maps--see section 2) have produced very promising results. Details, including an analysis of error bounds, will appear elsewhere. I have also tried alternate versions where the patches are constructed as low-order polynomials which are best approximations in the ℓ_1 or ℓ_∞ sense (via linear programming) again with adaptive choice of data. The results were very similar. Finally, I have also experimented with computing patch approximations, followed by continuous least-squares tensor-product spline approximation. Again, the experiments were promising.

8. Contouring

As indicated in the introduction, frequently the goal in

fitting a surface f to data is to construct a contour map which approximates the contour map of the unknown surface F which produced the data. In this section we discuss some methods for constructing contour maps of a surface f.

8.1. Piecewise linear functions on triangles. When the function f to be contoured is a piecewise linear function defined on triangles (and globally continuous), locating contours reduces essentially to a matter of good bookkeeping. Indeed, if H is the height of the contour of interest, then it is easily seen that for a given triangle T with vertices, A, B, and C,

(8.1) the contour does not pass through T if $H < \min(f(A), f(B), f(C))$ or if $H > \max(f(A), f(B), f(C))$

and

(8.2) the contour intersects exactly two sides of T otherwise.

If case (8.2) holds, it is easy to determine which two sides are intersected and, moreover, by using inverse linear interpolation between vertex values, the points on these sides where the contour crosses can be determined. Specifically, if, for example,

$$f(A) < H < f(B),$$

then the contour crosses the line from A to B at the point on the line which is a distance of

$$\frac{(H-f(A))}{(f(B)-f(A))} \, |B - A|$$

from A. Given the points on two sides of a triangle where the contour line crosses, we can now draw the contour line since it is simply a straight line between the points. An algorithm to carry out this procedure requires enumerating the triangles and vertices and some kind of effective search procedure. There are several available in the literature. For ALGOL programs,

see Heap [111,112]. (An earlier paper of Heap and Pink [113] contains a similar FORTRAN program but only for regular triangularizations.) Lawson [128] discusses a similar algorithm. The algorithms mentioned include two possible approaches: (1) one may start with a triangle where it is known the contour intersects, and trace this contour as far as it goes, or (2) one may simply draw the contour lines in all triangles which have them.

8.2. Piecewise bilinear functions on rectangles. Suppose now that the function f to be contoured is a piecewise (continuous) function on a rectangle partitioned into subrectangles by a grid. Since f is linear in x or y on the edges, it follows that we can again determine whether a contour line of height H crosses an edge by inverse linear interpolation. There is in this case, however, a serious difficulty which does not arise in the case of triangles. It may happen that the height H lies on three or even four sides of the rectangle. In this case, it is possible that two different contour lines pass through the rectangle, and it is not clear how to interconnect the points (see Figure 6).

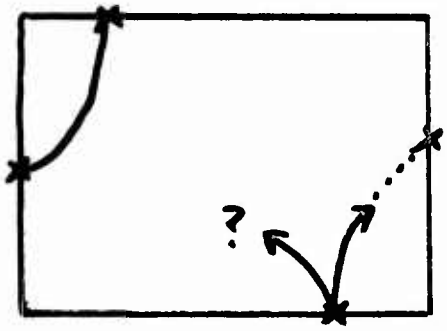


Figure 6. Two Contours in a Rectangle

Put another way, if we are following a contour and enter a rectangle as shown above in Figure 6 on the bottom line, then it is not clear whether we should now turn right or turn left. One approach to designing an algorithm in this case is to simply

always go right, say, even though this may in the end be wrong. (If it is, we have to start over with a coarser mesh.) This technique was incorporated in an algorithm by Heap [111,112]--the paper contains a FORTRAN program. (An earlier ALGOL program can be found in Heap and Pink [113].

A second approach to handling the ambiguity problem is compute an approximation to the value of f at the center of the rectangle (e.g., by taking the average of the four-corner values) and then to triangulate the rectangle. This amounts to a second-stage approximation process, and the surface contoured is no longer f itself but an approximation g. This idea was programmed in ALGOL in Heap and Pink [113] and in FORTRAN in Heap [111,112].

Once the set of points for a particular contour have been found, there are a variety of ways of drawing a contour line through these points. One possibility is to simply draw straight lines between each of the points. The actual contour lines are expressions of the form $y = (a+bx)/(c+dx)$ in each rectangle. These are generally not straight lines. Hence, if smoother contours are desired, one may use any one of a number of methods for drawing a smooth curve through an ordered set of points in the plane. For example, the curve could be computed in parametric form using one-dimensional splines. Another possibility would be to use the Bezier methods with either Bernstein polynomials or with B-splines (cf. Gordon and Riesenfeld [89] and Riesenfeld [163]), although in this case the curves will not exactly go through the points. For other algorithms see Marlow and Powell [138] or McConalogue [139].

8.3. Piecewise quadratics on triangles. Suppose now that f is a piecewise quadratic defined on a triangular partition. In this case a contour line at height H passing through a triangle must be described by a conic section. Such a section can

be represented in parametric form as

$$x(t) = (b_0 + b_1 t + b_2 t^2)/(b_3 + b_4 t + b_5 t^2)$$

$$y(t) = (b_6 + b_7 t + b_8 t^2)/(b_3 + b_4 t + b_5 t^2),$$

see Powell [156]. Powell has promised an algorithm based on this observation.

We turn now to some methods for handling general functions f on arbitrary domains D.

8.4. A simple line-printer method. The following simple-minded method can produce reasonable-looking contours without excessive computation, and without recourse to a plotter. Suppose H is a rectangle enclosing the domain D, and that we partition H as $H = \cup H_{ij}$ with a rectangular grid as in (6.2). Let $HL < HU$ be given real numbers. Finally, suppose that t_{ij} is some point in H_{ij} where f can be evaluated (perhaps one of the corners or the center). Define

$$(8.3)\quad C_{ij} = \begin{cases} 0 & , \text{ if } f(t_{ij}) < HL \\ 9 & , \text{ if } f(t_{ij}) > HU \\ v \text{ , if } HL + (v-1)h < f(t_{ij}) < HL + vh, & 1 \le v \le 8, \end{cases}$$

for all $i = 0,1,\ldots,k$ and $j = 0,1,\ldots,l$ (where $h = (HU-HL)/8$). The $(k+2)$ by $(\ell+2)$ matrix C contains only integers, and if it is printed out without either horizontal or vertical spacing, we obtain a reasonable-looking contour map of the function. A typical example is included in Figure 7. The method can be refined by using an alpha-numeric array C and more than 10 symbols. It can also be refined by using a printer with appropriate horizontal spacing so that each symbol occupies a square rather than a rectangle (e.g., cf. Buneman [40]).

8.5. Threading on a rectangular grid. As in section 8.4,

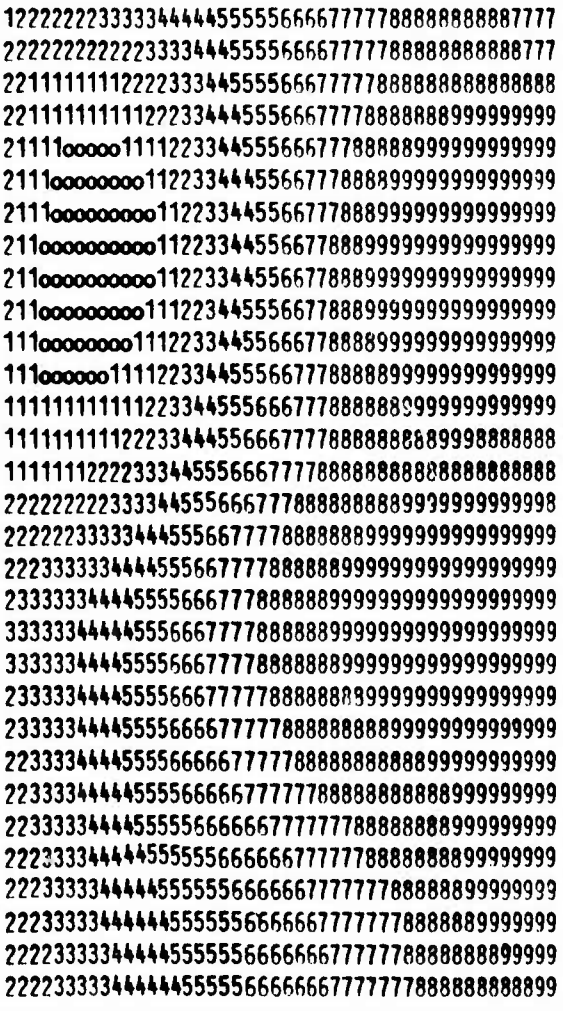


Figure 7. A Simple Contour Map (Heart Potential)

suppose that D is imbedded in a rectangle H which has been partitioned by a rectangular grid as in (6.2). Assuming that f is continuous, it is still possible to decide which of the grid lines a particular contour of height H crosses by examining the end-points of each such line. Since f is not generally linear along such a line, we cannot determine exactly where the crossing point is by linear inverse interpolation. However, if we are willing to evaluate f a few times along this line, we can estimate the crossing point quite accurately by bisection,

for example. Once a sequence of points on a contour has been determined, we may thread a curve through the points just as in section 8.2.

This method does have one serious drawback, however,-- just as with the method discussed in section 8.2--, if we are tracing a contour it may happen that after entering a triangle there is an ambiguity as to which of two points to use to exit the rectangle. One could opt for an ad hoc rule or try the second-stage approximation described in section 8.2. For an example of how this method works, see Falconer [76] (based on Lodwick and Whittle [130]), where it is applied to a surface constructed by local weighted quadratic polynomial least-squares approximation. Since bisection is involved, one should realize that in drawing contours with this routing the surface f is going to be evaluated a great many times.

8.6. Threading on a triangular grid. An obvious cure for the ambiguity discussed in section 8.5 for threading on a rectangular grid is to use a triangular partition in the first place. Then the bisection method coupled with a threading routine leads immediately to a contouring routine for general surfaces f. Strangely enough, I have not been able to find anywhere where this method has been suggested.

I have made no effort to track down all available papers on contouring. A few which I did find and have not yet mentioned are Cottafawa and le Moli [60], Dayhoff [64], and Pelto et al [152]. There are many others.

In some cases it may be desirable to have a more graphic picture of a surface than a contour map can provide. Recently there has been considerable effort devoted to computer methods for displaying surfaces on a scope or with a plotter. For some examples of output and a discussion of methods, see e.g. the book by Barnhill and Riesenfeld [20] on computer-aided design.

If an actual 3-D picture is desired instead of just a perspective, it is even possible to produce holographs.

References

1. Ahlberg, J. H., E. N. Nilson, and J. L. Walsh, The Theory of Splines and Their Applications, Academic Press, New York, 1967.
2. Ahlberg, J. H., E. N. Nilson, and J. L. Walsh, Extremal, orthogonality, and convergence properties of multidimensional splines, J. Math. Anal. Appl. 11 (1965), 27-48.
3. Ahlin, A. C., A bivariate generalization of Hermite's interpolation formula, Math. Comp. 18 (1964), 264-273.
4. Ahuja, D. V., and S. A. Coons, Interactive graphics in data processing, geometry for construction and display, IBM Systems J. 7 (1968), 188-205.
5. Akima, H., Bivariate interpolation and smooth surface fitting based on local procedures, Comm. ACM. 17 (1974), 26-31.
6. Akima, H., A method of bivariate interpolation and smooth surface fitting based on local procedures, Comm. ACM 17 (1974), 18-20.
7. Anderson, W. L., Application of bicubic spline functions to two-dimensional gridded data, U.S. Geological Survey Rpt. GD-71-022, 1971.
8. Arthur, D. W., Multivariate spline functions, I. Construction, properties, and computation, J. Approximation Theory 12 (1974), 396-411.
9. Arthur, D. W., Multivariate spline functions, II. Best error bounds, J. Approximation Theory 15 (1975), 1-10.
10. Atteia, M., Étude de certains noyaux et théorie des fonctions (spline) en Analysis Numérique, thèse, Grenoble, 1966.
11. Atteia, M., Existence et détermination des fonctions splines a plusieurs variables, C. R. Acad. Sci. Paris 262 (1966), 575-578.
12. Atteia, M., Fonctions (spline) et noyaux reproduisants D' Aronszajn-Bergman, Rev. Francaise Informat, Rechereche Operationnelle 4E Annee, R-3 (1970), 31-43.
13. Aziz, A. K. (ed.), The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations, Academic Press, New York, 1972.

14. Babuska, I., Approximation by ill functions, Comment.Math. Univ. Carolinae 11 (1970), no. 4, 787-811.

15. Barnhill, R. E., Smooth interpolation over triangles, in [20], 45-70.

16. Barnhill, R. E., Birkhoff, G., and W. J. Gordon, Smooth interpolation in triangles, J. Approximation Theory 8 (1973), 114-128.

17. Barnhill, R. E., and J. A. Gregory, Compatible smooth interpolation in triangles, J. Approximation Theory 15 (1975), 214-225.

18. Barnhill, R. E., and J. A. Gregory, Blending function interpolation to boundary data on triangles, TR 14, Depart. Math., Brunel Univ., 1972.

19. Barnhill, R. E., and L. Mansfield, Error bounds for smooth interpolation in triangles, J. Approximation Theory 11 (1974), 306-318.

20. Barnhill, R. E., and R. F. Riesenfeld, (eds.), Computer Aided Geometric Design, Academic Press, New York, 1974.

21. Bhattacharyya, B. K., An automatic method of computation and mapping of high resolution aeromagnetic data, Geophysics 36 (1971), 695-716.

22. Bhattacharyya, B. K., Bicubic spline interpolation as a method for treatment of potential field data, Geophysics 34 (1969), 402-423.

23. Bhattacharryya, B. K., and B. Raychaudhuri, Aeromagnetic and geologic interpretation of a section of the Appalachian belt in Canada, Canad. J. Earth Sci. 4 (1967), 1015-1037.

24. Biersack, J. P., and D. Fink, Channeling, blocking, and range measurements using thermal neutron induced reactions, in Atomic Collisions in Solids, Vol. 2, edited by S. Datz, B. R. Appleton, and C. D. Moak, Plenum Publishing Corp., New York, 1973, 737-747.

25. Birkhoff, Garrett, Piecewise bicubic interpolation and approximation in polygons, in Approximation with Special Emphasis on Spline Functions, edited by I. J. Schoenberg, Academic Press, New York, 1969, 185-222.

26. Birkhoff, G., and de Boor, C., Piecewise polynomial interpolation and approximation, in Approximation of Functions, H. L. Garabedian, ed., Elsevier, Amsterdam, 1965, 164-190.

27. Birkhoff, G., and H. Garabedian, Smooth surface interpolation, J. Math. Phys. 39 (1960), 258-268.

28. Birkhoff, G., and L. Mansfield, Compatible triangular finite elements, J. Math. Anal. Appl. 47 (1974), 531-553.

29. Birkhoff, G., M. H. Schultz, and R. S. Varga, Piecewise Hermite interpolation in one and two variables with applications to partial differential equations, Numer. Math. 11 (1968), 232-256.

30. Böhmer, K., G. Meinardus, and W. Schempp (eds.), Spline Functions, Lecture Notes 501, Springer-Verlag, New York, 1976.

31. Bohrenberg, G., and E. Giese, Statistical Methoden und ihre Anwedungen in der Geographie, Studienbuch der Geographie, 1975.

32. de Boor, C., Bicubic spline interpolation, J. Math. and Phys. 41 (1962), 212-218.

33. de Boor, Carl, On calculating with B-splines, J. Approximation Theory 6 (1972), 50-62.

34. de Boor, Carl, Appendix to splines and histograms by I. J. Schoenberg, in Spline Functions and Approximation Theory, A. Meir and A. Sharma, eds., ISNM 21, Birkhauser, 1973, 329-358.

35. de Boor, Carl (ed.), Mathematical Aspects of Finite Elements in Partial Differential Equations, Academic Press, New York, 1974.

36. de Boor, Carl, Splines as linear combinations of B-splines, a survey, this volume.

37. de Boor, C., and G. J. Fix, Spline approximation by quasi-interpolants, J. Approximation Theory 8 (1973), 19-45.

38. de Boor, C., T. Lyche, and L. Schumaker, On calculating with B-splines, II. Integration, to appear Proc. Oberwolfach, 1975.

39. Bramble, J., and M. Zlamal, Triangular elements in the finite element method, Math. Comp. 24 (1970), 809-820.

40. Buneman, O., Sub-grid resolution of flow and force fields, Report no. 452, Inst. for Plasma Research, Stanford Univ., Stanford, California, 1972.

41. Cadwell, J. H., A least squares surface-fitting program, Comput. J. 3 (1961), 266-269.

42. Cadwell, J. H., and D. E. Williams, Some orthogonal methods of curve and surface fitting, Comput. J. 4 (1961), 260.

43. Call, E. S., and F. F. Judd, Surface fitting by separation, J. Approximation Theory 12 (1974), 283-290.

44. Carlson, R. E., and C. A. Hall, Bicubic spline interpolation and approximation in right triangles, J. Math. of Comp., to appear.

45. Carlson, R. E., and C. A. Hall, Ritz approximations to two-dimensional boundary-value problems, Numer. Math. 18 (1971), 171-181.

46. Carlson, R. E., and C. A. Hall, Bicubic spline interpolation in rectangular polygons, J. Approximation Theory 6 (1972), 366-377.

47. Carlson, R. E., and C. A. Hall, Error bounds for bicubic spline interpolation, J. Approximation Theory 7 (1973), 41-47.

48. Carlson, R. E., and C. A. Hall, On piecewise polynomial interpolation in rectangular polygons, J. Approximation Theory 4 (1971), 37-53.

49. Carlson, R. E., and C. A. Hall, Piecewise polynomial approximation of smooth functions in rectangular polygons, J. Approximation Theory 4 (1971), 37-53.

50. Cavendish, J. C., Automatic triangulation of arbitrary planar domains for the finite element method, Int. J. for Numer. Methods in Engineering 8 (1974), 679-696.

51. Chorley, R. J., ed., Spatial Analysis in Geomorphology, Methuen and Co., London, 1972.

52. Ciarlet, P. G., and P. A. Raviart, General Lagrange and Hermite interpolation in $\mathbb{R}^N$ with applications to finite element methods, Arch. Rational Mech. Anal. 46 (1972), 177-199.

53. Ciarlet, P. G., and P. A. Raviart, Interpolation de Lagrange dans R^n, C. R. Acad. Sci. Paris Ser. A273 (1971), 578-581.

54. Ciarlet, P. G., and P. A. Raviart, Interpolation de Lagrange sur des eléments finis courbes dans $\mathbb{R}^n$, C. R. Acad. Sci. Paris Ser. A274 (1972), 640-643.

55. Ciarlet, P. G., and P. A. Raviart, Interpolation theory over curved elements with applications to finite element

methods, Comp. Meth. Appl. Mech. Eng. 1 (1972), 217-249.

56. Clenshaw, C. W., and J. G. Hayes, Curve and surface fitting, J. Inst. Math. Appl. 1 (1965), 164-183.

57. Coatmelec, M. C., Approximation et interpolation des fonctions differentiables de plusieurs variables, Ann. Scient. EC, Norm, Sup. 3, Serie 8 3 (1966), 271-341.

58. Collatz, L., Einschliessungssatz fur die Minimalabweichung bei der Segmentapproximation, Simposio Internazionale Sulle Applicazione dell Analisi alla Fisica Matematica (Cagliari-Sassari, 1964), 11-21.

59. Coons, S. A., Surface patches and B-spline curves, in [20], 1-16.

60. Cottafava, G., and G. Le Moli, Automatic contour map, Comm. ACM 12 (1969), 386-391.

61. Crain, I. K., and B. K. Bhattacharyya, Treatment of non-equispaced two-dimensional data with a digital computer, Geoexploration 5 (1967), 173-194.

62. Davis, J. C., Statistics and Data Analysis in Geology, John Wiley, New York, 1973.

63. Davis, P. J., Interpolation and Approximation, Blaisdell, New York, 1963.

64. Dayhoff, M. O., A contour map program of X-ray crystallography, Comm. ACM 6 (1963), 620-622.

65. Delvos, Franz-Jurgen, On surface interpolation, J. Approximation Theory 15 (1975), 209-213.

66. Delvos, F. J., and H. W. Kösters, On the variational characterization of bivariate interpolation methods, Math. Z. 145 (1975), 129-137.

67. Delvos, F. J., and G. Malinka, Das Blending-Schema von Spline Systemen, in Spline Functionen, K. Bohmer, G. Meinardus, and W. Schempp, eds., Bibliographisches Institut, Zurich, 1974, 47-58.

68. Delvos, F. J., and W. Schempp, Sard's method and the theory of spline systems, J. Approximation Theory 14 (1975), 230-243.

69. Delvos, F. J., and W. Schempp, An extension of Sard's method, in [30], 80-91.

70. Delvos, F. J., and K. H. Schlosser, Das Tensorproduktschema von Spline Systemen, in Splinefunktionen, K. Bohmer, G. Meinardus, and W. Schempp, eds., Bibliographisches Institut, Zurich, 1974, 59-74.

71. Doty, D. R., Blending function techniques with applications to discrete least squares, dissertation, Michigan State Univ., 1975.

72. Duchon, J., Fonctions-spline du type plaque mince en dimension 2, RPT, 231, Univ. of Grenoble, 1975.

73. Duchon, J., Fonctions-spline a energie invariante par rotation, Rpt. 27, Univ. of Grenoble, 1976.

74. Earnshaw, J. L., and I. M. Yuille, A method of fitting parametric equations for curves and surfaces to sets of points defining them approximately, Computer Aided Design 3 (1971), 19-22.

75. Efroysom, M. A., Multiple regression analysis, in Mathematical Methods for Digital Computers, A. Ralston and H. Wilff, eds., Wiley, New York, 1960, 191-203.

76. Falconer, K. J., A general purpose algorithm for contouring over scattered data points, NPL DNAC Rpt. 6, 1971.

77. Ferguson, J., Multivariable curve interpolation, J. Assoc. Comput. Mach. 11 (1964), 221-228.

78. Fisher, S. D., and J. W. Jerome, Elliptic variabional problems in L^2 and L^∞, Indiana J. Math. 23 (1974), 685-698.

79. Fisher, S. D., and J. W. Jerome, Spline solutions to L^1 extremal problems in one and several variables, J. Approximation Theory 13 (1975), 73-83.

80. Fix, George, and G. Strang, Fourier analysis of the finite element method in Ritz-Galerkin theory, Studies in Appl. Math. 48 (1969), 265-273.

81. Frederickson, Paul O., Triangular spline interpolation, Rpt. 6-70, Lakehead Univ, 1970.

82. Frederickson, P. O., Quasi-interpolation, extrapolation, and approximation on the plane, Conf. Numerical Maths., Winnipeg (1971), 159-167.

83. Göel, J. J., Construction of basic functions for numerical utilization of Ritz's method, Numer. Math. 12 (1968), 435-447.

84. Gordon, W. J., Blending-function methods of bivariate and multivariate interpolation and approximation, SIAM J. Numer. Anal. 8 (1971), 158-177.

85. Gordon, W. J., Spline-blended surface interpolation through curve networks, J. Math. Mech. 18 (1969), 931-952.

86. Gordon, W. J., Distributive lattices and the approximation of multivariate functions, in Approximation with Special

Emphasis on Spline Functions, I. J. Schoenberg, ed., Academic Press, New York, 1969, 223-277.

87. Gordon, W. J., Free-form surface interpolation through curve networks, General Motors Research Report GMR-921, 1969.

88. Gordon, W. J., and C. A. Hall, Transfinite element methods. Blending-function interpolation over arbitrary curved element domains, Numer. Math. 21 (1973), 109-129.

89. Gordon, W. J., and R. F. Riesenfeld, B-spline curves and surfaces, in [20], 95-126.

90. Gordon, W. J., and J. Wixom, A note on Shepard's method of metric interpolation to scattered bivariate and multivariate data, to appear.

91. Grant, F., A problem in the analysis of geophysical data, Geophysics 22 (1957), 309-344.

92. Greville, T. N. E., Note on fitting of functions of several variables, SIAM J. Appl. Math. 9 (1961), 109-115.

93. Guenther, R. B., and E. L. Roetman, Some observations on interpolation in higher dimensions, Math. Comp. 24 (1970), 517-522.

94. Hall, C. A., Bicubic interpolation over triangles, J. Math. Mech. 19 (1969), 1-11.

95. Hall, C. A., Natural cubic and bicubic spline interpolation, SIAM J. Numer. Anal. 10 (1973), 1055-1060.

96. Halliday, J., J. F. Wall, and W. D. Joyner, Report on multivariable curve fitting using fundamental splines, Report No. MSN.167, British Aircraft, 1972.

97. Hanson, R. J., J. R. Radbill, and C. L. Lawson, Write-up for SURF, CONT1, SPLVR1, Jet Propulsion Lab., Pasadena, Calif., 1972.

98. Harbaugh, J. W., and D. F. Merriam, Computer Applications in Stratigraphic Analysis, John Wiley, New York, 1968.

99. Haussman, Werner, Mehrdimensionale Hermite-interpolation in Iterationsverfahren, Numerische Mathematik Approximations Theorie, ISNM Vol. 15, L. Collatz, G. Meinardus, H. Unger, and H. Werner, eds., Birkhauser, Basel, 1970, 147-160.

100. Haussman, Werner, On multivariate spline systems, J. Approximation Theory 11 (1974), 285-305.

101. Haussmann, Werner, Tensorproduktmethoden bei mehrdimensionale Interpolation, Math. Z. 124 (1972), 191-198.

102. Haussmann, Werner, Tensorprodukte und mehrdimensionale Interpolation, Math. Z. 113 (1970), 17-23.

103. Haussmann, W., and H. B. Knopp, Divergenz und Konvergenz zweidimensionaler Lagrange-interpolation, to appear Z. Angew. Math. Mech.

104. Haussmann, Werner, and Heinz Josef Munch, On the construction of multivariate spline systems, in Approximation Theory, G. G. Lorentz, ed., Academic Press, New York, 1973, 373-378.

105. Hayes, J. G., Fitting data in more than one variable, in [106], 84-97.

106. Hayes, J. G., ed., Numerical Approximation to Functions and Data, The Athlone Press, London, 1970.

107. Hayes, J. G., New shapes from bicubic splines, NPLNAC 58, also Proceedings of CAD 74: International Conference in Engineering and Building Design, Fiche 36, Row G, International Publishing Co., London, 1974.

108. Hayes, J. G., Available algorithms for curve and surface fitting, in Software for Numerical Mathematics, D. J. Evans, ed., Academic Press, London, 1974, Chapter 12.

109. Hayes, J. G., Numerical methods for curve and surface fitting, Bull. Inst. Math. Applics. (1974), 144- .

110. Hayes, J. G., and J. Halliday, The least squares fitting of cubic spline surfaces to general data sets, J. Inst. Maths. Applics. 14 (1974), 89-103.

111. Heap, B. R., Algorithms for the production of contour maps over an irregular triangular mesh, NPL Rpt. NAC 10, 1972.

112. Heap, B. R., Two Fortran contouring routines, NPL NAC 47, 1974.

113. Heap, B. R., and M. G. Pink, Three contouring algorithms, NPL Math. Rpt. 81, 1969.

114. Hessing, R., H. K. Lee, A. Pierce, and E. N. Powers, Automatic contouring using bicubic functions, Geophysics 37 (1972), 669-674.

115. Holroyd, M. T., and B. K. Bhattacharyya, Automatic contouring of geophysical data using bicubic spline interpolation, Geol. Survey Canada Paper No. 70-55, 1970.

116. Hosaka, M., Theory of curve and surface synthesis and their smooth fitting, Information Processing in Japan 9 (1969), 60-68.

117. IMSL Library, International Mathematics and Statistics Library, Houston, Texas.

118. Jentzsch, G., G. Lange, and O. Rosenbach, Anwendung der Spline-funktionen zur Bearbeitung geopnysikalischer Messreihen, in [20], 99-115.

119. Jordan, T. L., Smoothing and multivariable interpolation with splines, LA 3137, Los Alamos, 1964.

120. Kaufman, E. H., and G. D. Taylor, Uniform rational approximation of functions of several variables, Int. J. for Numer. Methods in Engineering 9 (1975), 297-323.

121. Koelling, M. E. V., and E. H. T. Whitten, Fortran IV program for spline surface interpolation and contour map production, Geocomprograms 9 (1973), 1-12.

122. Krumbein, W. C., Trend surface analysis of contour-type maps with irregular control-point spacing, J. Geophysical Res. 64 (1959), 823-834.

123. Kubik, K., Approximation of measured data by piecewise bicubic polynomial functions, manuscript.

124. Kubik, K., The estimation of the weights of measured quantities within the method of least squares, Bull. Geodesique 95 (1970), 21-40.

125. Kubik, K., B. Kunji, and J. Kure, A computer programme for height block adjustment, ITC Rpt., Delft, 1968.

126. Kunz, K. S., Numerical Analysis, McGraw-Hill, New York, 1957.

127. Laurent, P. J., Approximation et Optimisation, Hermann, Paris, 1972.

128. Lawson, C., Generation of a triangular grid with applications to contour plotting, T.M. 299, JPL, 1972.

129. Lawson, C., and R. Hanson, Solving Least Squares Problems, Prentice-Hall, Englewood Cliffs, N.J., 1974.

130. Lodwick, G. D., and J. Whittle, A technique for automatic contouring field survey data, Austral. Comput. J. 2 (1970), 104-109.

131. Lyche, T., and L. Schumaker, Local spline approximation methods, J. Approximation Theory 15 (1975), 294-325.

132. Mangeron, D., Sopra un problema al contorno..., Rend. Accad. Sci. Fis. Mat. Napoli 2 (1932), 28-40.

133. Mansfield, L. E., On the optimal approximation of linear functionals in spaces of bivariate functions, SIAM J. Numer. Anal. 8 (1971), 115-126.

134. Mansfield, L. E., On the variational approach to defining splines on L-shaped regions, J. Approximation Theory 12 (1974), 99-112.

135. Mansfield, L. E., On the variational characterization and convergence of bivariate splines, Numer. Math. 20 (1972), 99-114.

136. Mansfield, L. E., Optimal approximation and error bounds in spaces of bivariate functions, J. Approximation Theory 5 (1972), 77-96.

137. Mansfield, L. E., Interpolation to boundary data in triangles with application to compatible finite elements, this volume.

138. Marlow, S. M., and M. J. D. Powell, A FORTRAN subroutine for drawing a curve through a given sequence of data points, Report No. R 7092, A.E.R.E., Harwell, 1972.

139. McConalogue, D. J., An automatic French-curve procedure for use with an incremental plotter, Comput. J. 14 (1971), 207-209.

140. Merriam, D. F., ed., Computer Applications in the Earth Sciences, Plenum Press, New York, 1969.

141. Mitchell, A. R., Introduction to the mathematics of finite elements, in [198], 37-58.

142. Mitchell, A. R., and G. M. Phillips, Construction of basis functions in the finite element method, Nord, Tidskr, Inform. BIT 12 (1972), 81-89.

143. Munteanu, M. J., Generalized smoothing spline functions for operators, SIAM J. Numer. Anal. 10 (1973), 28-34.

144. Munteanu, M. J., On the construction of multidimensional splines, in Spline Functions and Approximation Theory, A. Meir and A. Sharma, eds., ISNM 21, Birkhauser, 1973, 235-265.

145. Munteanu, M. J., and L. L. Schumaker, Some multidimensional spline approximation methods, J. Approximation Theory 10 (1974), 23-40.

146. Nicolaides, R. A., On the class of finite elements generated by Lagrange interpolation, SIAM J. Numer. Anal. 9 (1972), 435-445.

147. Nicolaides, R. A., On a class of finite elements generated by Lagrange interpolation, II, SIAM. J. Numer. Anal. 10 (1973), 182-189.

148. Nielson, G. M., Bivariate spline functions and the approximation of linear functionals, Numer. Math. 21 (1973), 138-160.

149. Nielson, G. M., Multivariate smoothing and interpolating splines, SIAM J. Numer. Anal. 11 (1974), 435-446.

150. Nielson, G. M., Surface approximation and data smoothing using generalized spline functions, thesis, Univ. of Utah, 1970.

151. Norcliffe, G. B., On the use and limitations of trend surface models, Canadian Geographer 13 (1969), 338-348.

152. Pelto, C., T. Elkins, & H. Boyd, Automatic contouring of irregularly spaced data, Geophysics 33 (1968), 424-430.

153. Pilcher, D. T., Smooth parametric surfaces, in [20], 237-254.

154. Pivovarova, N. B., and T. B. Puknacheva, Smoothing experimental data with local splines, Sem. Num. Methods of Applied Mathematics, Novosibirsk, 1975.

155. Poeppelmeir, C., A Boolean sum interpolation scheme to random data for computer aided geometric design, thesis, Univ. of Utah, 1975.

156. Powell, M. J. D., Piecewise quadratic surface fitting for contour plotting, in Software for Numerical Analysis, D. J. Evans, ed., Academic Press, 1974, 253-272.

157. Prenter, P. M., Splines and Variational Methods, Wiley-Interscience, New York, 1975.

158. Prenter, P. M., Lagrange and Hermite interpolation in Banach spaces, J. Approximation Theory 4 (1971), 419-432.

159. Price, J. F., and R. H. Simonsen, Various methods and computer routines for approximation, curve fitting and interpolation, Boeing Rpt. D1-82-0151, 1962.

160. Rabinowitz, P., Applications of linear programming to numerical analysis, SIAM Rev. 10 (1968), 121-159.

161. Rabinowitz, P., Mathematical programming and approximation, in Approximation Theory, A. Talbot, ed., Academic Press, London, 1970, 217-231.

162. Reilly, W. I., Contouring gravity anomaly maps by digital plotter, New Zealand J. Geology Geophysics 12 (1969), 628-632.

163. Riesenfeld, R., Applications of B-spline approximation to geometric problems of computer-aided design, Ph.D. thesis, Univ. of Utah 1973.

164. Ritter, K., Two-dimensional splines and their extremal properties, Z. Angew. Math. Mech. 49 (1969), 597-608.

165. Ritter, K., Two-dimensional spline functions and best approximations of linear functionals, J. Approximation Theory 3 (1970), 352-368.

166. Robinson, J. E., H. A. K. Charlesworth, and M. J. Ellis, Structural analysis using spatial filtering in interior plans of South-Central Alberta, Amer. Assoc. Petrol. Geol. Bull. 53 (1969), 2341-2367.

167. Rosen, J. B., Minimum error bounds for multidimensional spline approximation, J. Comput. System Sci. 5 (1971), 430-452.

168. Salzer, H. E., Formulas for bivariate hyperosculatory interpolation, Math. Comp. 25 (1971), 119-133.

169. Salzer, H. E., Note on interpolation for a function of several variables, Bull. Amer. Math. Soc. 51 (1945), 279-280.

170. Salzer, H. E., Note on multivariate interpolation for unequally spaced arguments, with application to double summation, SIAM J. Appl. Math. 5 (1957), 254-262.

171. Sard, A., Approximation based on nonscalar observations, J. Approximation Theory 8 (1973), 315-334.

172. Sard, A., Instances of generalized splines, in Spline-Functionen, K. Bohmer, G. Meinardus, W. Schempp, eds., Bibliographisches Institut, Mannheim, 1974, 215-241.

173. Schaback, R., Konstruktion und algebraische Eigenschaften von M-spline Interpolierenden, Numer. Math. 21 (1973), 166-180.

174. Schaback, R., Kollokation mit mehrdimensionalen Splinefunktionen, in Numerische Behandlung nichtlinearer Integrodifferential und Differentialgleichungen, R.Ansorge, and W. Tornig, eds., Lecture Notes 395, Springer-Verlag, Heidelberg, 1974, 291-300.

175. Schlosser, K. H., Mehrdimensionale Spline-interpolation mit Hilfe der Methods von Sard, in [20], 247-264.

176. Schoenberg, I. J., Splines and histograms, in Spline Functions and Approximation Theory, A. Meir and A. Sharma, eds., ISNM 21, Birkhauser, 1973, 277-327.

177. Schultz, M. H., L-infinity-multivariate approximation theory, SIAM J. Numer. Anal. 6 (1969), 161-183.

178. Schultz, M. H., Multivariate L-spline interpolation, J. Approximation Theory 2 (1969), 127-135.

179. Schultz, M. H., Spline Analysis, Prentice-Hall, Englewood Cliffs, N. J., 1973.

180. Shepard, D., A two-dimensional interpolation function for irregularly spaced data, Proc. 1964 ACM Nat. Conf., 517-524.

181. Shu, H., S. Hori, W. R. Mann, and R. N. Little, The synthesis of sculptured surfaces, in Numerical Control Programming Languages, Proc. First Intern. IFIP/IFAC Proclamation Conf., Rome, 1969, North-Holland, Amsterdam, 1970, 358-375.

182. Sommer, M., Alternanten bei gleichmässiger Approximation mit zweidimensional Splinefunktionen, in [20], 339-370.

183. Späth, H., Algorithmus 10- zweidimensionale glatte interpolation, Computing 4 (1969), 178-182.

184. Späth, H., Two-dimensional exponential splines, Computing 7 (1971), 364-369.

185. Späth, H., Two-dimensional glatte interpolation, Computing 4 (1969), 178-182.

186. Steffensen, J. F., Interpolation, Williams and Wilkins, Baltimore, 1927.

187. Stewart, G. W., Introduction to Matrix Computations, Academic Press, New York, 1973.

188. Strang, G., and J. Fix, An Analysis of the Finite Element Method, Prentice-Hall, Englewood Cliffs, N. J., 1973.

189. Thacher, H. C., Jr., Derivation of interpolation formulas in several independent variables, Annal. of the New York Acad. of Sciences 86 (1960), 758-775.

190. Thacher, H. C., Jr., and W. E. Milne, Interpolation in several variables, SIAM J. Appl. Math. 8 (1960), 33-42.

191. Theilheimer, F., and W. Starkweather, The fairing of ship lines on a high-speed computer, Math. Comp. 15 (1961), 338-355.

192. Thomann, Jean, Détermination et construction de fonctions spline à deux variables définies sur un domaine rectangulaire ou circulaire, thesis, Lille, 1970.

193. Thomann, J., Obtention de la fonction spline D interpolation A 2 variables sur un domaine rectangulaire ou circulaire, in Proc. Algol en Analyse Numerique II, Centre National de la Recherche Scientifique, Paris, 1970, 83-94.

194. Wachspress, E. L., A rational basis for function approximation, Part 2, J. Inst. Math. Appl. 11 (1971), 83-104.

195. Wachspress, E. L., A Rational Finite Element Basis, Academic Press, New York, 1975.

196. Weinstein, S. E., Approximation of functions of several variables: product Chebychev approximations, J. Approximation Theory 2 (1969), 433-447.

197. Whaples, G. W., A note on degree N independence, SIAM J. Appl. Math. 6 (1958), 300-301.

198. Whiteman, J. R. (ed.), The Mathematics of Finite Elements, Academic Press, London, 1973.

199. Whiten, W. J., The use of multi-dimensional cubic spline functions for regression and smoothing, Austral. Comput. J. 3 (1971), 81-88.

200. Whitten, E. H. T., Scalar and directional field and analytical data for spatial variability studies, Math. Geol. 6 (1974), 183-198.

201. Whitten, E. H. T., Orthogonal polynomial trend surfaces for irregularly spaced data, Int. Assoc. Math. Geol. J. 2 (1970), 141-152.

202. Whitten, E. H. T., More on irregularly spaced data and orthogonal polynomial trend surfaces, Int. Assoc. Math. Geol. J. 4 (1972), 83.

203. Whitten, E. H. T., The practical use of trend-surface analyses in the geological sciences, in Display and Analysis of Spatial Data, Davis and McCullagh, eds., John Wiley, New York, 282-297.

204. Whitten, E. H. T., Orthogonal polynomial contoured trend-surface maps for irregularly spaced data, Tr. 2, Geology Dept., Northwestern Univ., 1974.

205. Whitten, E. H. T., Trends in computer applications in structural geology, in Computer Applications in the Earth Sciences, Plenum Press, New York, 1969, 233-249.

206. Whitten, E. H. T., and M. E. V. Koelling, Computation of Bicubic-spline surfaces for irregularly-spaced data, Tr. 3, Geology Dept., Northwestern Univ., 1974.

207. Whitten, E. T. H., and M. E. V. Koelling, Geological use of multidimensional spline functions, Math. Geology 5 (1973).

208. Whitten, E. H. T., and M. E. V. Koelling, Spline surface interpolation, spatial filtering, and trend surfaces for geologically mapped variables, Math. Geol. 5 (1973), 111-126.

209. Zavialov, Yu.S., An optimal property of bicubic spline functions and the problem of smoothing (Russian), Vycisl, Sistemy, 42 (1970), 109-158.

210. Zavialov, Yu. S., Interpolating L-splines in several variables, Mat. Zametki, 14 (1973), 11-20.

211. Zavialov, Yu. S., Smoothing L-splines in several variables, Mat. Zametki, 15 (1974), 371-379.

212. Zavialov, Yu. S., L-spline functions of several variables, Dokl. Akad. Nauk, SSSR 214 (1974), also Soviet Math. Dokl. 15 (1974), 338-341.

213. Zenisek, A., Interpolation polynomials on the triangle, Numer. Math. 15 (1970), 283-296.

214. Zienkiewicz, O. C., The finite element method: from intuition to generality, Appl.Mech.Rev. 23 (1970),249-256.

215. Zlamal, M., A finite element procedure of the second order of accuracy, Numer. Math. 14 (1970), 394-402.

216. Zlamal, M., On the finite element method, Numer. Math. 12 (1968), 394-409.

217. Zlamal, M., Curved elements in the finite element method I, II, SIAM J. Numer. Anal. 10 (1973), 229-240; 11 (1974) 347-362.

218. Zwart, Philip B., Multi-variate splines with non-degenerate partitions, SIAM J. Numer. Anal. 10 (1973), 665-673.

Larry L. Schumaker
Department of Mathematics and Center for Numerical Analysis
The University of Texas at Austin, 78712.

Supported in part by Grant AFOSR 74-2598B.